The **lwarp** package

LaTeX to HTML

v0.60 — 2018/09/19

© 2016–2018 Brian Dunn
bd@BDTechConcepts.com

Abstract

The **lwarp** package allows LaTeX to directly produce HTML5 output, using external utility programs only for the final conversion of text and images. Math may be represented by SVG files or MathJax.

Documents may be produced by pdfLaTeX, LuaLaTeX, or XeLaTeX. A **texlua** script removes the need for system utilities such as **make** and **gawk**, and also supports **xindy** and **latexmk**. Configuration is automatic at the first manual compile.

Print and HTML versions of each document may coexist, each with its own set of auxiliary files. Support files are self-generated on request. Assistance is provided for import into EPUB conversion software and word processors.

A modular package-loading system uses the **lwarp** version of a package for HTML when available. More than 300 LaTeX packages are supported with these high-level source-compatibility replacements, and many others work as-is.

A tutorial is provided to quickly introduce the user to the major components of the package.

To update existing projects, see section 2: Updates.

For a list of supported features, see table 1: Supported packages and features.

Note that this is still a "beta" version of lwarp, and some things may change in response to user feedback and further project development.

License:

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 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 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

1 Supporting TEX development

TEX and related projects:

- are mostly open-sourced and a volunteer effort;
- benefit students, academics, scientists, engineers, and businesses;
- help drive education, public and private research, and commercial activity;
- are used in the fields of mathematics, science, engineering, and humanities;
- span decades of development;
- are enduring — many older packages are still actively used and maintained;
- are largely backwards compatible;
- are portable across all the major computing platforms;
- are usable even on older computers and away from internet access;
- are continuing to maintain relevance with modern improvements;
- require no yearly subscription fees;
- and are supported by an active community of knowledgeable volunteers.

Please consider helping by joining and/or contributing to the TEX Users Group, a United States 501(c)(3) tax-exempt charitable organization. Contributions are accepted by credit card, check, or Pay Pal, via the United Way, or by USA or European bank transfer. Membership in TUG supports the development of TEXLive, the major TEX distribution.

Donations may be directed towards individual projects:

TUG Bursary Fund: Assistance for attending annual TUG meetings.
CTAN: The Comprehensive TEX Archive Network — Central storage for TEX.
TeX Development Fund: Support for specific projects.
EduTeX: Teaching and using TEX in schools and universities.
GUST e-foundry fonts: Enhanced for math and additional language groups.
LaTeX Project: Modernizing the LATEX core.
Libre Font Fund: Fonts, tools (FontForge), and distribution (the Open Font Library).
LuaTeX: Combining the pdfTEX engine and the Lua language.
MetaPost: Postscript graphics.
MacTeX: TEX for Mac.
PDF Accessibility: Modern PDF standards.
Other: Additional projects may be specified.

To make a contribution: https://www.tug.org/donate.html

For users of MiKTEX: https://miktex.org/donations.html

For country-specific TEX users groups: http://tug.org/usergroups.html

Contents

1 Supporting TEX development 2

List of Figures 49

List of Tables 49

2 Updates 50

3 Introduction 67
3.1 Supported packages and features . . . 69

4 Alternatives 74
4.1 Internet class . . . 74
4.2 TeX4ht . . . 74
4.3 Translators . . . 74
4.4 AsciiDoc and Asciidoctor . . . 75
4.4.1 Asciidoctor-LaTeX . . . 75
4.5 Pandoc . . . 75
4.6 Word processors . . . 75
4.7 Commercial systems . . . 75
4.8 Comparisons . . . 75

5 Installation 77
5.1 Installing the **lwarp** package . . . 79
5.2 Installing the **lwarpmk** utility . . . 81
5.2.1 Using a local copy of **lwarpmk** . . . 82
5.3 Installing additional utilities . . . 83

6 Tutorial 85
6.1 Starting a new project . . . 85
6.2 Compiling the print version with **lwarpmk** . . . 89
6.3 Compiling the HTML version with **lwarpmk** . . . 90
6.4 Generating the SVG images . . . 91
6.5 Using MathJax for math . . . 92
6.6 Changing the CSS style . . . 93
6.7 Customizing the HTML output . . . 93
6.8 Using **latexmk** . . . 94
6.9 Using XeLaTeX or LuaLaTeX . . . 95
6.10 Using DVI LaTeX . . . 95
6.11 Using a glossary . . . 96
6.12 Cleaning auxiliary files . . . 97
6.13 Cleaning auxiliary and output files . . . 97
6.14 Cleaning the images from the `lateximages` directory . . . 97
6.15 Converting PDF or EPS images to SVG . . . 97
6.16 Creating HTML from an incomplete compile . . . 97

6.17 Processing multiple projects in the same directory . . . 98
6.18 Using the **make** utility . . . 98

7 Converting an existing document 99

8 Additional details 100
8.1 Shell escape . . . 100
8.2 Font and UTF-8 support . . . 100
8.2.1 Indexes, glossaries, and encoding . . . 101
8.3 **lwarp** package loading and options . . . 102
8.4 Customizing the HTML output . . . 106
8.4.1 Example HTML file naming . . . 111
8.5 Customizing the CSS . . . 112
8.6 Selecting the operating system . . . 113
8.7 Selecting actions for print or HTML output . . . 114
8.8 Commands to be placed into the `warpprint` environment . . . 115
8.9 Title page . . . 115
8.10 HTML page meta descriptions . . . 116
8.11 HTML page meta title . . . 117
8.12 HTML page meta author . . . 117

9 Special cases and limitations 117
9.1 Things to avoid . . . 118
9.2 Formatting . . . 118
9.2.1 Text formatting . . . 118
9.2.2 Horizontal space . . . 118
9.2.3 Text alignment . . . 119
9.2.4 Accents . . . 119
9.2.5 **textcomp** package . . . 119
9.2.6 Superscripts and other non-math uses of math mode . . . 119
9.2.7 Empty `\item` followed by a new line of text or a nested list: . . . 119
9.2.8 Filenames and URLs in lists or footnotes . . . 120
9.2.9 **relsize** package . . . 120
9.3 Boxes and minipages . . . 120
9.3.1 Marginpars . . . 120
9.3.2 Save Boxes . . . 120
9.3.3 Minipages . . . 121
9.3.4 Side-by-side minipages . . . 121
9.3.5 Framed minipages and other environments . . . 121
9.3.6 **fancybox** package . . . 123
9.3.7 **mdframed** package . . . 124
9.4 Cross-references . . . 125
9.4.1 Page references . . . 125
9.4.2 **cleveref** and **varioref** packages . . . 125
9.4.3 Hyperlinks, **hyperref**, and **url** . . . 125
9.4.4 Footnotes and page notes . . . 126

9.5 Front and back matter . . . 127
9.5.1 Custom classes with multiple authors and affiliations . . . 127
9.5.2 Starred chapters and sections . . . 127
9.5.3 **abstract** package . . . 128
9.5.4 **titling** and **authblk** . . . 128
9.5.5 **tocloft** package . . . 128
9.5.6 **appendix** package . . . 128
9.5.7 **pagenote** package . . . 129
9.5.8 **endnotes** package . . . 129
9.5.9 BibTeX . . . 129
9.5.10 **glossaries** package . . . 129
9.5.11 Indexing overview . . . 130
9.5.12 Indexing with basic LaTeX and **makeidx** . . . 131
9.5.13 Indexing with **index** . . . 132
9.5.14 Indexing with **splitidx** . . . 133
9.5.15 Indexing with **imakeidx** . . . 135
9.5.16 Indexes with **memoir** . . . 138
9.5.17 Using a custom **makeindex** style file . . . 141
9.5.18 Using a custom **xindy** style file . . . 142
9.5.19 Additional indexing limitations . . . 143
9.5.20 Index positions, TOC, **tocbibind** . . . 143
9.6 Math . . . 144
9.6.1 Rendering tradeoffs . . . 144
9.6.2 SVG option . . . 145
9.6.3 MathJax option . . . 146
9.6.4 Customizing MathJax . . . 146
9.6.5 MathJax limitations . . . 146
9.6.6 Catcode changes . . . 147
9.6.7 Dynamic math . . . 148
9.6.8 Display math . . . 148
9.6.9 **chemformula** package . . . 148
9.6.10 **mhchem** package . . . 148
9.6.11 **ntheorem** package . . . 149
9.6.12 **siunitx** package . . . 149
9.6.13 **units** and **nicefrac** packages . . . 149
9.6.14 **newtxmath** package . . . 149
9.7 Graphics . . . 150
9.7.1 **tikz** package . . . 151
9.7.2 **grffile** package . . . 151
9.7.3 **color** package . . . 151
9.7.4 **xcolor** package . . . 152
9.7.5 **epstopdf** package . . . 152
9.7.6 **pstricks** package . . . 152
9.7.7 **pdftricks** package . . . 152
9.7.8 **overpic** package . . . 153
9.8 Tabbing . . . 153

9.9 Tabular . . . 153
9.9.1 **longtable** package . . . 156
9.9.2 **supertabular** and **xtab** packages . . . 157
9.9.3 **bigdelim** package . . . 157
9.10 Floats . . . 157
9.10.1 Float contents alignment . . . 157
9.10.2 **float**, **trivfloat**, and/or **algorithmicx** together . . . 158
9.10.3 **caption** and **subcaption** packages . . . 158
9.10.4 **subfig** package . . . 158
9.10.5 **floatrow** package . . . 158
9.10.6 **keyfloat** package . . . 159
9.11 Koma-Script . . . 159
9.12 Memoir . . . 159
9.13 Miscellaneous . . . 160
9.13.1 **verse** and **memoir** . . . 160
9.13.2 **newclude** package . . . 161
9.13.3 **babel** package . . . 161
9.13.4 **polyglossia** package . . . 161
9.13.5 **todonotes** and **luatodonotes** packages . . . 162
9.13.6 **fixme** . . . 162
9.13.7 xparse . . . 162

10 EPUB conversion 163

11 Word-processor conversion 165
11.1 Activating word-processor conversion . . . 165
11.2 Additional modifications . . . 167
11.3 Recommendations . . . 168
11.4 Limitations . . . 169

12 Modifying lwarp 171
12.1 Creating a development system . . . 171
12.2 Modifying a package for **lwarp** . . . 173
12.2.1 Adding a package to the **lwarp.dtx** file . . . 174
12.3 Modifying a class for **lwarp** . . . 174
12.4 Testing **lwarp** . . . 174
12.5 Modifying **lwarpmk** . . . 175

13 Troubleshooting 176
13.1 Using the **lwarp.sty** package . . . 176
13.1.1 Debug tracing output . . . 180
13.2 Compiling the `lwarp.dtx` file . . . 180

1 lwarp.sty 181

14 Implementation 181

15 Section depths and HTML headings 182

16 Source Code 183

17 Detecting the TEX Engine — pdflatex, lualatex, xelatex 184

18 MD5 hashing 184

19 pdfLaTeX T1 and UTF8 encoding 185

20 Unicode input characters 185

21 Miscellaneous tools 186

22 Early package requirements 186

23 Operating-System portability 187
23.1 Common portability code . . . 187
23.2 Unix, Linux, and Mac OS . . . 187
23.3 MS-WINDOWS . . . 188

24 Package options 188
24.1 Conditional compilation . . . 193

25 Package load order 195
25.1 Tests of package load order . . . 195
25.2 Error for disallowed packages loaded before lwarp . . . 197
25.3 Enforcing package loading after lwarp . . . 197

26 Required packages 204

27 Loading packages 208

28 Additional required packages 211

29 File handles 211

30 Include a file 212

31 Copying a file 213

32 Debugging messages 214

33 Defining print and HTML versions of macros and environments 215

34 HTML-conversion output modifications 219
34.1 User-level controls . . . 220

34.2 Heading adjustments 221

35 Remembering original formatting macros 223

36 Accents 225

37 Configuration Files 227
37.1 project_html.tex 227
37.2 **lwarpmk** configuration files 227
37.2.1 lwarpmk.conf 234
37.2.2 <project>.lwarpmkconf 234
37.3 lwarp.css 234
37.4 lwarp_sagebrush.css 264
37.5 lwarp_formal.css 269
37.6 sample_project.css 273
37.7 lwarp.ist 274
37.8 lwarp.xdy 275
37.9 lwarp_one_limage.cmd 275
37.10 lwarp_mathjax.txt 276
37.11 lwarpmk.lua — lwarpmk option 278

38 Stacks 297
38.1 Assigning depths 297
38.2 Closing actions 298
38.3 Closing depths 298
38.4 Pushing and popping the stack 299

39 Data arrays 300

40 Localizing catcodes 301

41 Localizing dynamic math 302

42 Sanitizing labels and filenames 303

43 HTML entities 304

44 HTML filename generation 304

45 Homepage link 307

46 `\LWRPrintStack` diagnostic tool 308

47 Closing stack levels 308

48 PDF pages and styles 309

49 HTML tags, spans, divs, elements 310

49.1 Mapping LaTeX Sections to HTML Sections . . . 311
49.2 Babel-French tag modifications . . . 311
49.3 HTML tags . . . 312
49.4 Block tags and comments . . . 314
49.5 Div class and element class . . . 315
49.6 Single-line elements . . . 316
49.7 HTML5 semantic elements . . . 317
49.8 High-level block and inline classes . . . 317
49.9 Closing HTML tags . . . 319

50 Paragraph handling 320

51 Paragraph start/stop handling 324

52 Page headers and footers 327

53 CSS 327

54 Title, HTML meta author, HTML meta description 328

55 Footnotes 329
55.1 Regular page footnotes . . . 330
55.2 Minipage footnotes . . . 330
55.3 Titlepage thanks . . . 330
55.4 Regular page footnote implementation . . . 330
55.5 Minipage footnote implementation . . . 333
55.6 Printing pending footnotes . . . 334

56 Marginpars 335

57 Splitting HTML files 336

58 Sectioning 344
58.1 User-level starred section commands . . . 345
58.2 Book class commands . . . 345
58.3 Sectioning support macros . . . 346
58.4 `\section` and friends . . . 353

59 Starting a new file 355

60 Starting HTML output 358

61 Ending HTML output 361

62 Title page 363
62.1 Setting the title, etc. . . . 363
62.2 `\if@titlepage` . . . 365
62.3 Changes for `\affiliation` . . . 365

62.4 Printing the thanks . . . 366
62.5 Printing the title, etc. in HTML . . . 366
62.6 Printing the title, etc. in print form . . . 368
62.7 \maketitle for HTML output . . . 368
62.8 \published and \subtitle . . . 372

63 Abstract 373

64 Quote and verse 374
64.1 Attributions . . . 374
64.2 Quotes, quotations . . . 374
64.3 Verse . . . 375
64.3.1 LaTeX core verse environment . . . 375
64.3.2 **verse** and **memoir** . . . 376

65 Verbatim and tabbing 376

66 Theorems 379

67 Lists 380
67.1 List environment . . . 381
67.2 Itemize . . . 385
67.3 Enumerate . . . 385
67.4 Description . . . 386
67.5 Patching the lists . . . 386

68 Tabular 388
68.1 Limitations . . . 388
68.2 Temporary package-related macros . . . 391
68.2.1 **arydshln** . . . 391
68.3 Token lookahead . . . 391
68.4 Tabular variables . . . 392
68.5 Handling &, @, !, and bar . . . 395
68.5.1 Handling & . . . 397
68.5.2 Filling an unfinished row . . . 398
68.6 Handling \\ . . . 400
68.7 Looking ahead in the column specifications . . . 401
68.8 Parsing @, >, <, !, bar columns . . . 402
68.9 Parsing 'l', 'c', or 'r' columns . . . 406
68.10 Parsing 'p', 'm', or 'b' columns . . . 406
68.11 Parsing 'w' columns . . . 407
68.12 Parsing 'D' columns . . . 407
68.13 Parsing the column specifications . . . 407
68.14 **colortbl** and **xparse** tabular color support . . . 412
68.15 Starting a new row . . . 414
68.16 Printing vertical bar tags . . . 415
68.17 Printing at or bang tags . . . 415

68.18 Data opening tag . . . 416
68.19 Midrules . . . 418
68.20 Cell colors . . . 425
68.21 Multicolumns . . . 429
68.21.1 Parsing multicolumns . . . 429
68.21.2 Multicolumn factored code . . . 433
68.21.3 Multicolumn . . . 436
68.21.4 Longtable captions . . . 436
68.21.5 Counting HTML tabular columns . . . 439
68.22 Multirow if not loaded . . . 441
68.23 Multicolumnrow . . . 441
68.24 Utility macros inside a table . . . 442
68.25 Special-case tabular markers . . . 443
68.26 Checking for a new table cell . . . 444
68.27 `\mrowcell` . . . 447
68.28 `\mcolrowcell` . . . 447
68.29 HTML `tabular` environment . . . 448

69 Cross-references 454
69.1 Setup . . . 454
69.2 Zref setup . . . 456
69.3 Labels . . . 457
69.4 References . . . 459
69.5 Hyper-references . . . 462

70 Floats 466
70.1 Float environment . . . 467
70.2 Float tracking . . . 469
70.3 Caption inside a float environment . . . 470
70.4 Caption and LOF linking and tracking . . . 471

71 Table of Contents, LOF, LOT 474
71.1 Reading and printing the TOC . . . 475
71.2 High-level TOC commands . . . 478
71.3 Side TOC . . . 478
71.4 Low-level TOC line formatting . . . 480

72 Index and glossary 483

73 Bibliography presentation 486

74 Restoring original formatting 487

75 Math 489
75.1 Limitations . . . 489
75.1.1 Rendering tradeoffs . . . 489
75.1.2 SVG option . . . 490

75.1.3 MathJax option . . . 490
75.1.4 Customizing MathJax . . . 491
75.1.5 MathJax limitations . . . 491
75.1.6 Catcode changes . . . 492
75.1.7 Dynamic math . . . 492
75.1.8 Display math . . . 493
75.2 HTML `alt` tag names . . . 493
75.3 Inline and display math . . . 493
75.4 MathJax support . . . 505
75.5 Equation environment . . . 507
75.6 `\displaymathnormal` and `\displaymathother` . . . 510
75.7 AMS Math environments . . . 512
75.7.1 Support macros . . . 512
75.7.2 Environment patches . . . 513

76 Lateximages 516
76.1 Description . . . 516
76.2 Support counters and macros . . . 517
76.3 Font size . . . 518
76.4 Sanitizing math expressions for HTML . . . 518
76.5 Equation numbers . . . 520
76.6 HTML `alt` tags . . . 521
76.7 `lateximage` environment . . . 522

77 `center`, `flushleft`, `flushright` 528

78 Pre-loaded packages 530

79 Siunitx 530

80 Graphics print-mode modifications 531
80.1 General limitations . . . 531
80.2 Print-mode modifications . . . 533

81 Xcolor boxes 533

82 Chemmacros environments 537

83 Cleveref 537

84 Picture 541

85 Boxes and Minipages 541
85.1 Counters and lengths . . . 542
85.2 Footnote handling . . . 542
85.3 Minipage handling . . . 543
85.4 Parbox, mbox, makebox, framebox, fbox, raisebox . . . 546

86 Direct formatting 552

87 Skips, spaces, font sizes 559

88 \phantomsection 567

89 \LaTeX and other logos 568
89.1 HTML logos 568
89.2 Print logos 570

90 \AtBeginDocument, \AtEndDocument 571

91 Koma-script 571

92 Memoir 571

93 Trademarks 572

2 lwarp-2up.sty 573

94 2up 573

3 lwarp-a4.sty 573

95 a4 573

4 lwarp-a4wide.sty 573

96 a4wide 573

5 lwarp-a5comb.sty 574

97 a5comb 574

6 lwarp-abstract.sty 574

98 abstract 574

7 lwarp-accsupp.sty 576

99 accsupp 576

8 lwarp-acro.sty 577

100 acro 577

9 lwarp-acronym.sty 578

101 acronym 578

10 lwarp-adjmulticol.sty 580

102 adjmulticol 580

11 lwarp-addlines.sty 581

103 addlines 581

12 lwarp-ae.sty 581

104 ae 581

13 lwarp-aecc.sty 581

105 aecc 581

14 lwarp-afterpage.sty 581

106 afterpage 581

15 lwarp-algorithm2e.sty 582

107 algorithm2e 582

16 lwarp-algorithmicx.sty 586

108 algorithmicx 586

17 lwarp-alltt.sty 587

109 alltt 587

18 lwarp-amsmath.sty 587

110 amsmath 587

19 lwarp-amsthm.sty 594

111 amsthm 594

20 lwarp-anonchap.sty 598

112 anonchap 598

21 lwarp-anysize.sty 599

113 anysize 599

22 lwarp-appendix.sty 599

114 appendix 599

23 lwarp-arabicfront.sty 600

115 arabicfront 600

24 lwarp-array.sty 600

116 array 600

25 lwarp-arydshln.sty 601

117 arydshln 601

26 lwarp-asymptote.sty 603

118 asymptote 603

27 lwarp-atbegshi.sty 603

119 atbegshi 603

28 lwarp-attachfile.sty 604

120 attachfile 604

29 lwarp-attachfile2.sty 605

121 attachfile2 605

30 lwarp-authblk.sty 607

122 authblk 607

31 lwarp-axessibility.sty 608

123 axessibility 608

32 lwarp-axodraw2.sty 609

124 axodraw2 609

33 lwarp-backref.sty 609

125 backref 609

34 lwarp-balance.sty 609

126 balance 609

35 lwarp-bibunits.sty 610

127 bibunits 610

36 lwarp-bigdelim.sty 610

128 bigdelim 610

37 lwarp-bigstrut.sty 611

129 bigstrut 611

38 lwarp-blowup.sty 612
130 blowup 612

39 lwarp-booklet.sty 612
131 booklet 612

40 lwarp-bookmark.sty 613
132 bookmark 613

41 lwarp-booktabs.sty 613
133 booktabs 613

42 lwarp-bophook.sty 615
134 bophook 615

43 lwarp-boxedminipage.sty 615
135 boxedminipage 615

44 lwarp-boxedminipage2e.sty 615
136 boxedminipage2e 615

45 lwarp-breakurl.sty 616
137 breakurl 616

46 lwarp-breqn.sty 616
138 breqn 616

47 lwarp-bxpapersize.sty 618
139 bxpapersize 618

48 lwarp-bytefield.sty 619
140 bytefield 619

49 lwarp-cancel.sty 619
141 cancel 619

50 lwarp-canoniclayout.sty 620
142 canoniclayout 620

51 lwarp-caption.sty 620
143 caption 620

52 lwarp-caption2.sty 623
144 caption2 623

53 lwarp-cases.sty 624
145 cases 624

54 lwarp-ccaption.sty 624
146 ccaption 624

55 lwarp-changebar.sty 625
147 changebar 625

56 lwarp-changepage.sty 625
148 changepage 625

57 lwarp-chngpage.sty 626

149 chngpage **626**

58 lwarp-chappg.sty 626

150 chappg **626**

59 lwarp-chapterbib.sty 626

151 chapterbib **626**

60 lwarp-chemfig.sty 627

152 chemfig **627**

61 lwarp-chemformula.sty 628

153 chemformula **628**

62 lwarp-chemgreek.sty 633

154 chemgreek **633**

63 lwarp-chemmacros.sty 634

155 chemmacros **634**
155.1 Changes to the user's document 634
155.2 Code 635
155.3 Loading modules 635
155.4 New environments 635
155.5 Acid-base 636
155.6 Charges 638
155.7 Nomenclature 638
155.8 Particles 640
155.9 Phases 641
155.10 Mechanisms 642
155.11 Newman 644
155.12 Orbital 645
155.13 Reactions 645
155.14 Redox 646
155.15 Scheme 647
155.16 Spectroscopy 647

155.17 Thermodynamics 652

64 lwarp-chemnum.sty 654

156 chemnum 654

65 lwarp-cite.sty 655

157 cite 655

66 lwarp-clrdblpg.sty 655

158 clrdblpg 655

67 lwarp-color.sty 655

159 color 655

68 lwarp-colortbl.sty 656

160 colortbl 656

69 lwarp-continue.sty 658

161 continue 658

70 lwarp-crop.sty 658

162 crop 658

71 lwarp-cuted.sty 658

163 cuted 658

72 lwarp-cutwin.sty 659

164 cutwin 659

73 lwarp-dblfloatfix.sty 660

165 dblfloatfix 660

74 lwarp-dblfnote.sty 660

166 dblfnote 660

75 lwarp-dcolumn.sty 660

167 dcolumn 660

76 lwarp-diagbox.sty 661

168 diagbox 661

77 lwarp-draftcopy.sty 662

169 draftcopy 662

78 lwarp-draftfigure.sty 663

170 draftfigure 663

79 lwarp-draftwatermark.sty 663

171 draftwatermark 663

80 lwarp-easy-todo.sty 664

172 easy-todo 664

81 lwarp-ebook.sty 665

173 ebook 665

82 lwarp-ellipsis.sty 665

174 ellipsis 665

83 lwarp-emptypage.sty 665

175 emptypage 665

84 lwarp-endfloat.sty 666

176 endfloat 666

85 lwarp-endheads.sty 666

177 endheads 666

86 lwarp-endnotes.sty 667

178 endnotes 667

87 lwarp-enumerate.sty 668

179 enumerate 668

88 lwarp-enumitem.sty 668

180 enumitem 668

89 lwarp-epigraph.sty 669

181 epigraph 669

90 lwarp-epstopdf.sty 670

182 epstopdf 670

91 lwarp-epstopdf-base.sty 670

183 epstopdf-base 670

92 lwarp-errata.sty 671

184 errata 671

93 lwarp-eso-pic.sty 673

185 eso-pic 673

94 lwarp-everypage.sty 674

186 everypage 674

95 lwarp-everyshi.sty 674

187 everyshi 674

96 lwarp-extramarks.sty 674

188 extramarks 674

97 lwarp-fancybox.sty 675

189 fancybox 675

98 lwarp-fancyheadings.sty 681

190 fancyheadings 681

99 lwarp-fancyhdr.sty 681

191 fancyhdr 681

100 lwarp-fancyref.sty 682

192 fancyref 682

101 lwarp-fancytabs.sty 683

193 fancytabs 683

102 lwarp-fancyvrb.sty 683

194 fancyvrb 683

103 lwarp-figcaps.sty 690

195 figcaps 690

104 lwarp-figsize.sty 690

196 figsize 690

105 lwarp-fix2col.sty 691

197 fix2col 691

106 lwarp-fixme.sty 691

198 fixme 691

107 lwarp-fixmetodonotes.sty 692

199 fixmetodonotes 692

108 lwarp-flafter.sty 693

200 flafter 693

109 lwarp-float.sty 693

201 float 693

110 lwarp-floatflt.sty 695

202 floatflt 695

111 lwarp-floatpag.sty 696

203 floatpag 696

112 lwarp-floatrow.sty 696

204 floatrow 696

113 lwarp-fltrace.sty 702

205 fltrace 702

114 lwarp-flushend.sty 702

206 flushend 702

115 lwarp-fnbreak.sty 703

207 fnbreak 703

116 lwarp-fncychap.sty 703

208 fncychap 703

117 lwarp-fnlineno.sty 704

209 fnlineno 704

118 lwarp-fnpos.sty 704

210 fnpos 704

119 lwarp-fontenc.sty 704

211 fontenc 704

120 lwarp-fontspec.sty 705

212 fontspec 705

121 lwarp-footmisc.sty 705

213 footmisc 705

122 lwarp-footnote.sty 706

214 footnote 706

123 lwarp-footnotehyper.sty 707

215 footnotehyper 707

124 lwarp-footnpag.sty 708

216 footnpag 708

125 lwarp-forest.sty 708

217 forest 708

126 lwarp-framed.sty 709

218 framed 709

127 lwarp-ftnright.sty 711

219 ftnright 711

128 lwarp-fullminipage.sty 711

220 fullminipage 711

129 lwarp-fullpage.sty 712

221 fullpage 712

130 lwarp-fullwidth.sty 712

222 fullwidth 712

131 lwarp-fwlw.sty 712

223 fwlw 712

132 lwarp-geometry.sty 713

224 geometry 713

133 lwarp-glossaries.sty 713

225 glossaries 713

134 lwarp-glossary.sty 716

226 glossary 716

135 lwarp-graphics.sty 716

227 graphics 716
227.1 Graphics extensions 716
227.2 Length conversions and graphics options 717
227.3 Printing HTML styles 719
227.4 `\includegraphics` 720
227.5 Boxes 725

136 lwarp-graphicx.sty 728

228 graphicx 728

137 lwarp-grffile.sty 729

229 grffile 729

138 lwarp-grid.sty 729

230 grid 729

139 lwarp-grid-system.sty 729

231 grid-system 729

140 lwarp-gridset.sty 730

232 gridset 730

141 lwarp-hang.sty 730

233 hang 730

142 lwarp-hanging.sty 732

234 hanging 732

143 lwarp-hypcap.sty 733

235 hypcap 733

144 lwarp-hypdestopt.sty 733

236 hypdestopt 733

145 lwarp-hypernat.sty 733

237 hypernat 733

146 lwarp-hyperref.sty 733

238 hyperref 733

147 lwarp-hyperxmp.sty 739

239 hyperxmp 739

148 lwarp-hyphenat.sty 740

240 hyphenat 740

149 lwarp-idxlayout.sty 741

241 idxlayout 741

150 lwarp-ifoddpage.sty 741

242 ifoddpage 741

151 lwarp-imakeidx.sty 742

243 imakeidx 742

152 lwarp-indentfirst.sty 747

244 indentfirst 747

153 lwarp-index.sty 747

245 index 747

154 lwarp-inputenc.sty 749

246 inputenc 749

155 lwarp-inputenx.sty 749

247 inputenx 749

156 lwarp-intopdf.sty 749

248 intopdf 749

157 lwarp-keyfloat.sty 750

249 keyfloat 750

158 lwarp-layaureo.sty 752

250 layaureo 752

159 lwarp-layout.sty 752

251 layout 752

160 lwarp-leading.sty 752

252 leading 752

161 lwarp-letterspace.sty 753

253 letterspace 753

162 lwarp-lettrine.sty 753

254 lettrine 753

163 lwarp-lineno.sty 754

255 lineno 754

164 lwarp-lips.sty 757

256 lips 757

165 lwarp-listings.sty 757

257 listings 757

166 lwarp-longtable.sty 762

258 longtable 762

167 lwarp-lscape.sty 764

259 lscape 764

168 lwarp-ltablex.sty 764

260 ltablex 764

169 lwarp-ltcaption.sty 765

261 ltcaption 765

170 lwarp-ltxgrid.sty 765

262 ltxgrid 765

171 lwarp-ltxtable.sty 765

263 ltxtable 765

172 lwarp-lua-check-hyphen.sty 766

264 lua-check-hyphen 766

173 lwarp-luacolor.sty 766

265 luacolor 766

174 lwarp-luatodonotes.sty 766

266 luatodonotes 766

175 lwarp-magaz.sty 769

267 magaz 769

176 lwarp-makeidx.sty 769

268 makeidx 769

177 lwarp-marginfit.sty 770

269 marginfit 770

178 lwarp-marginfix.sty 770

270 marginfix 770

179 lwarp-marginnote.sty 770

271 marginnote 770

180 lwarp-mcaption.sty 771

272 mcaption 771

181 lwarp-mdframed.sty 771

273 mdframed 771

273.1 Limitations . . . 772
273.2 Package loading . . . 772
273.3 Patches . . . 773
273.4 Initial setup . . . 773
273.5 Color and length HTML conversion . . . 774
273.6 Environment encapsulation . . . 774
273.7 Mdframed environment . . . 775
273.8 Titles and subtitles . . . 776
273.9 New environments . . . 779

182 lwarp-memhfixc.sty 782

274 memhfixc 782

183 lwarp-metalogo.sty 782

275 metalogo 782

184 lwarp-mhchem.sty 782

276 mhchem 782

185 lwarp-microtype.sty 785

277 microtype 785

186 lwarp-midfloat.sty 786

278 midfloat 786

187 lwarp-midpage.sty 786

279 midpage 786

188 lwarp-morefloats.sty 786

280 morefloats 786

189 lwarp-moreverb.sty 787

281 moreverb 787

190 lwarp-morewrites.sty 788

282 morewrites 788

191 lwarp-mparhack.sty 788

283 mparhack 788

192 lwarp-multicol.sty 789

284 multicol 789

193 lwarp-multirow.sty 790

285 multirow 790
285.1 Multirow 791
285.2 Combined multicolumn and multirow 792

194 lwarp-multitoc.sty 793

286 multitoc 793

195 lwarp-nameref.sty 794

287 nameref 794

196 lwarp-natbib.sty 794

288 natbib 794

197 lwarp-nccfancyhdr.sty 795

289 nccfancyhdr 795

198 lwarp-needspace.sty 796

290 needspace 796

199 lwarp-newclude.sty 796

291 newclude 796

200 lwarp-newunicodechar.sty 796

292 newunicodechar 796

201 lwarp-nextpage.sty 797

293 nextpage 797

202 lwarp-nicefrac.sty 797

294 nicefrac 797

203 lwarp-nonfloat.sty 798

295 nonfloat 798

204 lwarp-nonumonpart.sty 798

296 nonumonpart 798

205 lwarp-nopageno.sty 798

297 nopageno 798

206 lwarp-nowidow.sty 798

298 nowidow 798

207 lwarp-ntheorem.sty 799

299 ntheorem 799

299.1 Limitations . . . 799
299.2 Options . . . 800
299.3 Remembering the theorem style . . . 800
299.4 HTML cross-referencing . . . 804
299.5 `\newtheoremstyle` . . . 804
299.6 Standard styles . . . 804
299.7 Additional objects . . . 806
299.8 Renewed standard configuration . . . 806
299.9 **amsthm** option . . . 808
299.10 Ending a theorem . . . 810
299.11 `\NoEndMark` . . . 810
299.12 List-of . . . 810
299.13 Symbols . . . 811
299.14 Cross-referencing . . . 812

208 lwarp-overpic.sty 812

300 overpic 812

209 lwarp-pagegrid.sty 813

301 pagegrid 813

210 lwarp-pagenote.sty 813

302 pagenote 813

211 lwarp-pagesel.sty 813

303 pagesel 813

212 lwarp-paralist.sty 813

304 paralist 813

213 lwarp-parnotes.sty 814

305 parnotes 814

214 lwarp-parskip.sty 815
306 parskip 815
215 lwarp-pbox.sty 815
307 pbox 815
216 lwarp-pdflscape.sty 816
308 pdflscape 816
217 lwarp-pdfpages.sty 816
309 pdfpages 816
218 lwarp-pdfrender.sty 819
310 pdfrender 819
219 lwarp-pdfsync.sty 819
311 pdfsync 819
220 lwarp-pdftricks.sty 820
312 pdftricks 820
221 lwarp-pdfx.sty 820
313 pdfx 820
222 lwarp-pfnote.sty 821
314 pfnote 821
223 lwarp-phfqit.sty 821

315 phfqit 821

224 lwarp-placeins.sty 821

316 placeins 821

225 lwarp-prelim2e.sty 822

317 prelim2e 822

226 lwarp-prettyref.sty 822

318 prettyref 822

227 lwarp-preview.sty 822

319 preview 822

228 lwarp-pst-eps.sty 823

320 pst-eps 823

229 lwarp-pstricks.sty 823

321 pstricks 823

230 lwarp-quotchap.sty 824

322 quotchap 824

231 lwarp-quoting.sty 825

323 quoting 825

232 lwarp-ragged2e.sty 825

324 ragged2e 825

233 lwarp-realscripts.sty 826

325 realscripts 826

234 lwarp-register.sty 827

326 register 827

235 lwarp-relsize.sty 828

327 relsize 828

236 lwarp-repeatindex.sty 829

328 repeatindex 829

237 lwarp-resizegather.sty 830

329 resizegather 830

238 lwarp-romanbar.sty 830

330 romanbar 830

239 lwarp-romanbarpagenumber.sty 831

331 romanbarpagenumber 831

240 lwarp-rotating.sty 831

332 rotating 831

241 lwarp-rotfloat.sty 831

333 rotfloat 831

242 lwarp-savetrees.sty 832

334 savetrees 832

243 lwarp-scalefnt.sty 833

335 scalefnt 833

244 lwarp-schemata.sty 833

336 schemata 833

245 lwarp-scrextend.sty 834

337 scrextend 834

246 lwarp-scrhack.sty 837

338 scrhack 837

247 lwarp-scrlayer.sty 837

339 scrlayer 837

248 lwarp-scrlayer-notecolumn.sty 839

340 scrlayer-notecolumn 839

249 lwarp-scrlayer-scrpage.sty 839

341 scrlayer-scrpage 839

250 lwarp-section.sty 840

342 section 840

251 lwarp-sectionbreak.sty 841

343 sectionbreak 841

252 lwarp-sectsty.sty 842

344 sectsty 842

253 lwarp-setspace.sty 842

345 setspace 842

254 lwarp-shadow.sty 843

346 shadow 843

255 lwarp-showidx.sty 844

347 showidx 844

256 lwarp-showkeys.sty 844

348 showkeys 844

257 lwarp-sidecap.sty 845

349 sidecap 845

258 lwarp-sidenotes.sty 846

350 sidenotes 846

259 lwarp-SIunits.sty 847

351 SIunits 847

260 lwarp-siunitx.sty 848

352 siunitx 848

261 lwarp-soul.sty 853

353 soul 853

262 lwarp-soulpos.sty 855

354 soulpos 855

263 lwarp-soulutf8.sty 856

355 soulutf8 856

264 lwarp-splitidx.sty 856

356 splitidx 856

265 lwarp-stabular.sty 858

357 stabular 858

266 lwarp-stfloats.sty 859

358 stfloats 859

267 lwarp-subfig.sty 859

359 subfig 859

268 lwarp-subfigure.sty 864

360 subfigure 864

269 lwarp-supertabular.sty 865

361 supertabular 865

270 lwarp-syntonly.sty 866

362 syntonly 866

271 lwarp-t1enc.sty 867

363 t1enc 867

272 lwarp-tabls.sty 867

364 tabls 867

273 lwarp-tabularx.sty 867

365 tabularx 867

274 lwarp-tabulary.sty 868

366 tabulary 868

275 lwarp-textarea.sty 868

367 textarea 868

276 lwarp-textcomp.sty 869

368 textcomp 869
368.1 Limitations . . . 869
368.2 Package loading . . . 869
368.3 HTML symbols . . . 869
368.3.1 pdfLaTeX symbols . . . 869
368.3.2 XeLaTeX and LuaLaTeX symbols . . . 870
368.4 HTML dicritics . . . 871

277 lwarp-textfit.sty 872

369 textfit 872

278 lwarp-textpos.sty 872

370 textpos 872

279 lwarp-theorem.sty 873

371 theorem 873
371.1 Remembering the theorem style . . . 873
371.2 CSS patches . . . 875

280 lwarp-threeparttable.sty 877

372 threeparttable 877

281 lwarp-thumb.sty 878

373 thumb 878

282 lwarp-thumbs.sty 878

374 thumbs 878

283 lwarp-tikz.sty 879

375 tikz 879

284 lwarp-titleps.sty 880

376 titleps 880

285 lwarp-titleref.sty 884

377 titleref 884

286 lwarp-titlesec.sty 884

378 titlesec 884

287 lwarp-titletoc.sty 886

379 titletoc 886

288 lwarp-titling.sty 888

380 titling 888

289 lwarp-tocbasic.sty 893

381 tocbasic 893

290 lwarp-tocbibind.sty 893

382 tocbibind 893

291 lwarp-tocenter.sty 895

383 tocenter 895

292 lwarp-tocloft.sty 896

384 tocloft 896

293 lwarp-tocstyle.sty 902

385 tocstyle 902

294 lwarp-todo.sty 903

386 todo 903

295 lwarp-todonotes.sty 904

387 todonotes 904

296 lwarp-transparent.sty 906

388 transparent 906

297 lwarp-trimclip.sty 906

389 trimclip 906

298 lwarp-trivfloat.sty 907

390 trivfloat 907
390.1 Combining \newfloat, \trivfloat, and algorithmicx 908

299 lwarp-turnthepage.sty 908

391 turnthepage 908

300 lwarp-twoup-gen.sty 908

392 twoup-gen 908

301 lwarp-typearea.sty 909

393 typearea 909

302 lwarp-ulem.sty 909

394 ulem 909

303 lwarp-underscore.sty 912

395 underscore 912

304 lwarp-units.sty 912

396 units 912

305 lwarp-upref.sty 912

397 upref 912

306 lwarp-url.sty 912

398 url 912

307 lwarp-verse.sty 913

399 verse 913

308 lwarp-vertbars.sty 915

400 vertbars 915

309 lwarp-vmargin.sty 915

401 vmargin 915

310 lwarp-vowel.sty 916

402 vowel 916

311 lwarp-vwcol.sty 917

403 vwcol 917

312 lwarp-wallpaper.sty 919

404 wallpaper 919

313 lwarp-wasysym.sty 920

405 wasysym 920

314 lwarp-watermark.sty 920

406 watermark 920

315 lwarp-widows-and-orphans.sty 920

407 widows-and-orphans 920

316 lwarp-wrapfig.sty 921

408 wrapfig 921

317 lwarp-xcolor.sty 922

409 xcolor 922
409.1 Limitations . . . 922
409.2 Xcolor definitions: location and timing . . . 923
409.3 Package loading . . . 925
409.4 Remembering and restoring original definitions . . . 925
409.5 HTML color style . . . 925
409.6 HTML border . . . 926
409.7 High-level macros . . . 927
409.8 Row colors . . . 931

318 lwarp-xellipsis.sty 933

410 xellipsis 933

319 lwarp-xfrac.sty 934

411 xfrac 934

320 lwarp-xltabular.sty 937

412 xltabular 937

321 lwarp-xltxtra.sty 937

413 xltxtra 937

322 lwarp-xmpincl.sty 938

414 xmpincl 938

323 lwarp-xpiano.sty 938

415 xpiano 938

324 lwarp-xtab.sty 939

416 xtab 939

325 lwarp-xunicode.sty 941

417 xunicode 941

326 lwarp-xurl.sty 941

418 xurl 941

327 lwarp-xy.sty 941

419 xy 941

328 lwarp-zwpagelayout.sty 942

420 zwpagelayout 942

329 lwarp-patch-komascript.sty 944

421 patch-komascript 944

330 lwarp-patch-memoir.sty 946

422 patch-memoir 946

422.1 Packages . . . 947
422.2 Preliminary setup . . . 948
422.3 Laying out the page . . . 948
422.4 Text and fonts . . . 951
422.5 Titles . . . 951
422.6 Abstracts . . . 952
422.7 Docment divisions . . . 952
422.8 Pagination and headers . . . 955
422.9 Paragraphs and lists . . . 956
422.10 Contents lists . . . 957
422.11 Floats and captions . . . 961
422.12 Page notes . . . 965
422.13 Decorative text . . . 966
422.14 Poetry . . . 967
422.15 Boxes, verbatims and files . . . 967
422.16 Cross referencing . . . 968
422.17 Back matter . . . 969
422.18 Miscellaneous . . . 970
422.19 Ccaption emulation . . . 971
422.20 Final patchwork . . . 973

Change History and Index 974

List of Figures

1 tutorial.tex listing 86

List of Tables

1 LaTeX–HTML generation — lwarp package — Supported features . . . 69
2 Required software programs 78
3 Files created along with the print version 88
4 lwarp package options 103
5 HTML settings 107
6 Section HTML headings for word-processor conversion 169
7 Section depths and HTML headings 182
8 Tabular baseline 408
9 Tabular HTML column conversions 408
10 Cross-referencing data structures 455
11 Float data structures 466
12 AMSthm package — CSS styling of theorems and proofs 595
13 Ntheorem package — CSS styling of theorems and proofs 799
14 Theorem package — CSS styling of theorems and proofs 873

2 Updates

The following is a summary of updates to **lwarp**, highlighting new features and any special changes which must be made due to improvements or modifications in **lwarp** itself.

For a detailed list of the most recent changes, see the end of the Change History on page 997.

v0.60: Fixes for **longtable**, **listings**.

- **longtable**, etc.: Fixes for slowdown and memory management for very long tables.
- **listings**: Fix for HTML entities, and also when used inside a list.
- **diagbox**: Fix for incorrect HTML par tags.

packages
- Added **2up**, **twoup-gen**, **booklet**.
- Added **bophook**, **draftfigure**, **fullminipage**, **grid-system**, **layaureo**.
- Added **leading**, **widows-and-orphans**.
- Added **fancytabs**, **thumb**, **thumbs**.

v0.59: DVI **latex**, MATHJAX, **asymptote**, **pdftricks** and **pstricks**, **epstopdf**, **brqen**.

⚠ Reset the configuration
- Due to changes in **lwarpmk**, recompile any existing project a single time using ***pdflatex filename.tex*** or similar, after which **lwarpmk** may then be used with the new configuration files.

lwarpmk
- Added an error if `lwarpmk.conf`'s format has changed and the document must be recompiled.
- Added a warning if the `lwarpmk.conf` configuration file appears to be for the wrong operating system, in case files are transferred between systems.
- Added
  ```
  lwarpmk epstopdf <list-of-EPS-files>
  ```
  to quickly convert a document's EPS images to PDF or SVG. See section 9.7.

DVI **latex**
- Added support for DVI **latex**. See section 8.3.

latexmk
- Fix for `-shell-escape` with **latexmk**.

math
- Updated MathJax script to v2.7.4.
- Fix: Mathjax chapter number removed from non-numeric tagged equations.
- Added MathJax support for **nicefrac**, **units**.
- Fix for `\[` and `\]` with `\displaymathnormal`.

images
- Fix for `\includegraphics` filename expansion.
- `\includegraphics` now works with `.pdf` and `.eps` filename extensions.

packages
- Moved **amsmath** out of the **lwarp** core.

- Fix for **chemformula** `\NMR`.
- Added **asymptote**, **pdftricks**, **pstricks**, **pst-eps**.
- Added **breqn**, **SIunits**.
- Added **bxpapersize**, **canoniclayout**, **draftcopy**, **fnbreak**, **nccfancyhdr**.
- Added **accsupp**, **axessibility**.
- Added **xunicode**.
- Improved and now supports **epstopdf**.
- Tested to work as-is: **eepic**, **sepfootnotes**.

docs
- Added information about setting up a development version of **lwarp**.

v0.58: Extensive improvements in indexing, glossaries. Adds PDF-inclusion packages.

⚠ Reset the configuration
- Due to changes in **lwarpmk**, recompile any existing project a single time using ***pdflatex filename.tex*** or similar, after which **lwarpmk** may then be used with the new configuration files.

lwarpmk
- **lwarpmk**: Added the `-p` option to specify the project name.

glossaries
- **lwarpmk**: Now uses **makeglossaries** for glossary generation, allowing the processing of multiple glossaries at once.
- Added **lwarp** option `GlossaryCmd` to specify the shell command used by ***lwarpmk printglossary*** and ***lwarpmk htmlglossary***. Defaults to ***makeglossaries***.

index and glossary
- Docs: Extra indexing options. See section 9.5.11.
- Added support for **makeindex**. (Previously supported only **xindy**.) Also added indexing packages listed below.
- Added **lwarp** options `PrintIndexCmd`, `HTMLIndexCmd`, and `LatexmkIndexCmd` to specify shell commands used by ***lwarpmk printindex***, ***lwarpmk htmlindex***, and **latexmk**. May be preset with the `makeindex` or `xindy` **lwarp** options. See section 8.3.
- Added **lwarp** options `makeindex` and `xindy` to set `PrintIndexCmd`, `HTMLIndexCmd`, and `LatexmkIndexCmd` to sensible values for a typical single index. See section 8.3.
- Added **lwarp** option `makeindexStyle` to tell **lwarpmk** to use a custom style instead of `lwarp.ist`. See section 9.5.17.
- Fix for index entries with `\see`, `\seealso`, `\emph`, `\textbf`, etc.

misc. fixes
- Replaced each `\csuse` with `\@nameuse` for improved error detection.
- Additional internal print/HTML macro selection improvements.
- Fix: `\printindex` finishes pending `\index` writes first.

packages
- Fixes for **memoir**: **makeidx**, **ccaption**, multiple indexes, `\specialindex`.

- Fixes for **komascript**: Indexing improvements.
- Added **imakeidx**, **index**, **repeatindex**, **splitidx**.
- Added **attachfile**, **attachfile2**, **intopdf**, **pdfpages**, **pdfx**.
- Added **cases**.
- Tested to work as-is: **notes2bib**, **hvindex**.

v0.57: **algorithm2e**, **float** styles, tabular packages, internal improvements.

MathJax
- Added support for MathJax equations with `\footnote`, `\footnotemark`.

math macros
- Added `\StartDefiningMath` and `\StopDefiningMath` for use when defining macros in the preamble which contain $. See section 9.6.6.

dynamic math
- Added `\StartDynamicMath` and `\StopDynamicMath` to delimit math expressions which depend on a variable condition such as a counter. Such expressions will not be hashed for reuse, and will be converted to SVG math images even when MathJax is enabled. See section 9.6.7.

⚠ new name
- Renamed `\EndDefiningTabulars` to `\StopDefiningTabulars`.

`lateximage alt` tags
- Improved localization for `lateximage` HTML `alt` tags. For SVG math images, the `alt` tag under some conditions will be set to `\mathimagename`, which defaults to "math image". For packages, the `alt` tag is set using the package name followed by `\packagediagramname`, which defaults to "diagram". Ex:

  ```
  (-xy- diagram)
  ```

  See section 8.4.
- Fix: Improved print/HTML macro selection.

misc. fixes
- Fix: `\href` text catcodes.
- Fix: `\subref` text.
- Fixes: Colored `\rule` and `\boxframe`.

packages
- **float**, **rotfloat**: Adds support for float styles `ruled` and `boxed`.
- **float**: Fix: Do not create `\l@<type>` until `\listof` is used.
- **marginnote**: Fix: Long optional argument.
- **ellipsis**: Adds `\midwordellipsis`.
- **breakurl**: Fix for text catcodes.
- Added **algorithm2e**, **register**, **ltablex**, **xltabular**, **xellipsis**, **trimclip**, **errata**, **vowel**, **xpiano**.
- Prevents **glossary**.
- Tested to work as-is with **gauss**, **phonrule**, **piano**, **SIunits**, **tikzcodeblocks**.

v0.56: Shell escape, tabular packages.

lwarpmk

- Added
  ```
  lwarpmk pdftosvg <list-of-PDF-files>
  ```
  to quickly convert a document's PDF images to SVG, for use with HTML. See section 9.7.
- Added support for `-shell-escape`. See section 8.1.

`tabular`

- Added support for **array** `w` and `W` columns.
- Fix: `\multicolumn` parameter handling.
- Added support for double `\hlines`, `\midrules`, and vertical rules.
- Added support for **arydshln** dashed lines with HTML `tabular`, but reverts to plain rules for `lateximage` and SVG math `array`.

misc. fixes

- Fix: `\thinspace`.
- Fix: **paralist** compact environments.

packages

- Added **parnotes**, **quoting**, **lua-check-hyphen**, **tocenter**, **underscore**.
- Tested to work as-is with **babelbib**, **bibunits**, **bodegraph**, **fast-diagram**, **nicematrix**, **structmech**.

v0.55: Various fixes.

misc fixes

- Fix: Extraneous space in file links, which also prevented **Calibre** EPUB conversions.
- Fix: Float optional argument regression.
- Fix: `\ForceHTMLTOC` with `\phantomsection`.
- Fix: Overfull boxes in `lateximages`.
- Fix: QED symbols in `lateximage`.

packages

- **koma-script**: Fix: Figure with `\centering`, etc.
- Added **clrdblpg**.

v0.54: Float `\centering`, improved image checks.

⚠ Reset the configuration

- Due to changes in **lwarpmk**, recompile any existing project a single time using ***pdflatex filename.tex*** or similar, after which **lwarpmk** may then be used with the new configuration files.

lwarpmk

- `lwarpmk limages` checks for the presence of the HTML version of the document and valid image references before attempting to create the `lateximages`.
- **lwarpmk**: Improved error message if configuration file does not exist.

BibTeX

- Added documentation for avoiding error with BibTeX and `\etalchar`. See section 9.5.9.

polyglossia

- Added documentation regarding **polyglossia**. See section 9.13.4.

macros in section names

- Added documentation regarding the use of macros in section names. See section 9.1.

document encoding

⚠ New and revised encoding options

- Renamed and added package options:

Old Package Option	New Package Option
`xdyFilename`	`xindyStyle`
`IndexLanguage`	`xindyLanguage`
--	`xindyCodepage`
--	`pdftotextEnc`

  Use these options along with **inputenc** or **inputenx** to process documents in an encoding other than UTF-8. See section 8.2.

floats with `\centering`, etc.

- Floats now honor `\centering`, `\raggedright`, `\raggedleft`, and their **ragged2e** equivalents, when placed directly after:
  ```
  \begin{floattype}
  \centering
  ```

misc. fixes

- **tikz**: `\pgfpicture`, fit, align, font.
- **ragged2e**: `\centering` etc.
- **hyperref**: `\hypertarget` was creating duplicate of `\label`.
- **hyperref**: Active chars inside `\hyperref`, `\hyperlink`.
- **hyperref**: `\ref` inside `\hyperlink` caused a nested HTML link.
- **glossaries**: Fix when not using **babel** or **polyglossia**.
- **textcomp**: `\textperthousand`.
- LaTeX core `verse` environment: line spacing.

⚠

- Removed `\citetitle`, adjusted `\attribution`.

packages

- **memoir**: Minor update for v3.7g.
- Added **inputenx**, **bibunits**, **chngpage**, **forest**, **magaz**, **gridset**.
- Prevents loading **ae**, **aecc**, **t1enc**, and **wasysym**.

v0.53: Improved image checks.

lwarpmk

- **lwarpmk**: Added a warning about corrupted images due to the need to recompile the document one more time.
- **lwarpmk**: Added the ***`lwarpmk cleanlimages`*** command.
- Added documentation for ***`lwarpmk cleanlimages`*** and ***`lwarpmk pdftohtml`***.

v0.52: Improved footnotes, SVG math.

documentation

- Improved install instructions regarding `lwarp_baseline_marker.png`.
- Added documentation regarding footnotes in section headings, and footnotes with `\VerbatimFootnotes` from **fancybox**, **fancyvrb**. See section 9.4.4.

- Added documentation regarding font selection when using XeLaTeX or LuaLaTeX with **fontspec** and traditional font packages. See section 8.2.

SVG math
- Fix: Limit the number of background tasks when generating `lateximages`.
- Added user-adjustable SVG math font scaling. See section 76.3.
- Added warnings if `lwarp_baseline_marker.png` is not present, or if **graphicx** or **graphics** is not loaded.
- Improved `\ensuremath` hashing expansion.
- Fix: `equation*` with `split`.
- `tabbing` now works inside a `lateximage`. Use for math in `tabbing`.

MathJax
- Fix: MathJax script was not executing in some conditions.
- Added `\CustomizeMathJax` to add custom functions. See section 9.6.

footnotes
- Fix: Footnote numbering when using `HTMLDebugComments`.
- Fix: Footnote paragraph tags.
- Fix: `FootnoteDepth` defaults to `\subsubsection`.

misc. fixes
- Fix: `\kill` in a `lateximage`.
- Fix: `\FileDepth`, misc. others, when input encoding is not `utf8`.
- Fix: `\texorpdfstring` in a section name.

packages
- **hyperref** emulation: Fix for #, %, &, ~, _ characters in URLs.
- **fancybox**, **fancyvrb**: Initial support for `\VerbatimFootnotes`.
- **nicefrac**: Added with fix for `\ensuremath`.
- **graphicx**: Fix for option defaults. Added v1.1a/b options.
- **endfloat**: Updated for v2.6.
- **url**: Fixes for active characters.

v0.51: Improved SVG math, added numerous chemistry packages.

documentation
- Docs: Added Things to avoid.
- Docs: Added to Converting an existing document.
- Docs: Multiple authors and affiliations with custom classes. See section 9.5.1.
- Docs: **tikz** with matrices. See section 9.7.1.

SVG math
- Improved SVG math baseline.
- Improved SVG math font and color.
- Faster SVG math rendering.
- Improved support for display math containing complicated math objects, such as **tikz-cd**. See section 9.6.8.
- Fix: `\addcontentsline` inside SVG math.

- Fix: SVG math containing an embedded `lateximage`.

MathJax
- MathJax now handles `\ensuremath` in expressions.

misc. fixes
- Fix: Added `alignat` environment.
- Fix: **afterpackage** no longer required, which conflicted with **scrlfile**.
- Fix: **titling** `\thanks` mark.
- Fix: **fancybox** improvements.
- Fix: **tikz** `\tikz` macro. (Previously only the `tikzpicture` environment worked.)
- Fix: **tikz** with optional argument.

packages
- Added **mhchem**, **chemfig**, **chemformula**, **chemmacros**, **chemnum**, **chemgreek**, **epstopdf-base**, **grid**, **ltxgrid**.

v0.50: Improved SVG math.

SVG math
- SVG math and other `lateximages` now are converted to SVG using parallel background tasks, utilizing all available CPU cores.
- Inline SVG math image file names now are MD5 hashes made from their source LaTeX code. Identical inline math expressions, such as multiple instance of x, now share a single image file. This reduces the number of images to store, transmit, process, and display. Each image file is only converted to SVG a single time, and reused if it already exists. Display math and other forms of SVG image such as `picture` and Ti*k*z still use individual image files which are recreated each time ***`lwarpmk limages`*** is run.
- Fixes: SVG math and/or `\underline` in a sectioning file name.
- Improved SVG display math and tags.
- Improved SVG math and **siunitx** `alt` tags.
- Improved **siunitx** units.
- Fix: `\ensuremath` with MathJax now creates a `lateximage`.
- Fix: `\centering`, etc. in SVG math, `lateximage`, Ti*k*z.

misc. fixes
- Fix: Made various macros robust, additionally fixing **authblk**.
- Fix: **ntheorem** if neither `standard` nor `amsthm` selected.
- Fix: **listings**: Improved column alignment.
- Fix: Load **fontspec** if necessary.

packages
- Added **xy**, **epstopdf**, **diagbox**, **pbox**, **bytefield**, **axodraw2**, **phfqit**, **schemata**, **dblfloatfix**, **nonfloat**, **morefloats**.

v0.49:

tabular
- Added **xcolor** `\rowcolors`.
- Fix: `\noalign` inside a `tabular`.

math
- Fix: `\eqref` in a caption.

misc fixes
- Fix: Incorrect PDF font size changes caused occasional HTML corruption.
- Fix: **printlen** changes are now grouped for HTML output.

packages
- Added **vwcol**, **vertbars**, **hyphenat**, **lineno**, **fnlineno**, **figsize**, **hypdestopt**, **pagegrid**, **pdfrender**, **luacolor**, **resizegather**.

v0.48:

documentation
- Added some documentation regarding converting an existing document. See section 7.

cleveref
- Updated compatibility for new **cleveref** v0.21.

tabular
- Fix: Ignores optional tabular column arguments.

minor updates
- Added `\leftline`, `\centerline`, `\rightline`.
- Lists have improved font control via `\makelabel`.
- Print-mode `lateximage` now boxed to the natural width of its multiline contents.
- `abstract` now allows an optional name, as required by some classes.

math
- Fix: Improved spacing, `\mbox`, and font sizes with SVG math, Ti*k*z.
- **siunitx**: Improved SVG math, fraction compatibility, color output.

misc. fixes
- Fix: LOF/LOT links.
- Fix: Virtual page size grouping caused excessive PDF page breaks.
- Fix: Parsing similar package names in a single `\usepackage`.
- Fix: Adapts to classes without `\part`.
- Fix: `\newline` in `\title` was causing `<br>` in window title.
- Fix: `\maketitle` with `\cr`, `\crcr`, `\noalign`, for **IEEEtran** class.
- Fix: **xfrac** neutralized `BlockClass` and others.
- Fix: **todonotes** and **luatodonotes**: Improved `\todototoc`.

packages
- Added **colortbl**, **chapterbib**, **acro**, **acronym**, **hypernat**, **hypcap**, **stfloats**, **vmargin**, **fancyheadings**.
- **fancyref**: Now directly supported.

v0.47:

math
- Improved SVG math baseline and sizing.
- Fixes: SVGmath in captions, subcaptions, `\nameref`.
- Fixes: Line wrap at hyphen in HTML output.

packages
- Added **endheads**, **multitoc**, **sectionbreak**, **blowup**, **xurl**.

v0.46:

⚠ name change
- `\PrintStack` changed to `\LWRPrintStack`.

misc. fixes
- Fix: Empty lines between `tabular` rows.
- Fix: Stack unnesting.
- Fix: SVG math and `lateximages` in numerous situations.
- Fix: Spaces in `\usepackage`.
- Fix: Now allows MathJax inside `verse`.

v0.45:

documentation
- Improved **MikTeX** install instructions.
- Improved graphics and **epstopdf** instructions.
- Updates to the Introduction.

memoir
- Added **memoir**, **memhfixc**. See section 9.12.

cross-references
- Fix: Now allows underscores in labels.
- Fix: `\_` and `\<blank>` in section/file names.

math
- Fix: Now allows MathJax inside `tabbing`.

bibliography
- Fix: Bibliography `\em` names.
- Added **cite**, **natbib**, **backref**. (Also works as-is with **biblatex**.)

misc. fixes
- Fix: Empty lines between `tabular` rows.
- Fix: "Improper `\prevdepth`" with minipages, lists.
- Fix: Incorrect SVG math and `lateximages` with **subfig**.
- Fix: Lateximages from incorrect pages with Mathjax.
- Fix: Missing sidetoc if using **listings**.
- Fix: Added an **array** emulation package.

packages
- Added **subfigure**, **prettyref**, **hanging**, **midpage**, **flafter**, **fltrace**, **changebar**, **endfloat**, **continue**, **fwlw**, **turnthepage**, **footnpag**, **pagesel**, **textfit**, **titleref**.

v0.44:

koma-script
- Added **koma-script** classes (except **scrlttr2**, **scrjura**).
- Added **scrextend**, **scrlayer**, **scrlayer-notecolumn**, **scrlayer-scrpage**, **scrhack**, **tocstyle**, **tocbasic**.

HTML title and author
- Added `\HTMLTitle`. Fixed web page title if `\HTMLTitle` empty and no `\title` given and not using **titling** package.
- Fixed web page author if `\HTMLauthor` is empty and `\author` is not given.

encodings
- If using **pdflatex**, automatically loads T1 and UTF8 encodings. (Additional **fontenc** encodings may be loaded after **lwarp**.)

lists
- Added `list` and `trivlist` environments, **hang**.

tabular
- Fix: `\multicolumn` alignment if formatting for a word processor.
- Added **ltxtable**.

math
- Fix: MathJax combined with `lateximages`.
- **algorithmicx**: Improved comment symbol and floating.

packages
- Completed **todonotes** and **luatodonotes**.
- Added **todo**, **easy-todo**, **fixmetodonotes**, **fixme**.
- Added **soulutf8**, **soulpos**, **cancel**.
- Added **section**, **fancyref**, **ifoddpage**.
- Added **preview**, **atbegshi**, **watermark**.
- Improved **tocloft** `\newlistof` and `\newlistentry`.

v0.43:

- Docs: Reorganized HTML customization, added an HTML settings table. See section 8.4.

footnotes
- Added `FootnoteDepth` to control the placement of pending footnotes before section breaks. By default, pending footnotes are printed before each `\subparagraph` or higher.

sectioning
- Fix: Expansion in section name.

`tabular`
- Fix: Ignore spaces in tabular column specification.
- Fix: Tabular rules at bottom or when finishing incomplete rows.
- Fix: `\multicolumn` at/bang/before/after specifications, trim, and vertical rules.
- Fix: **supertabular** and **xtab** column misalignment.

math
- Fix: `equation*`.
- Fix: SVG math in a section name.
- Fix: `\ref` and `\eqref` in SVG math.

packages
- Added **todonotes** and **luatodonotes** (but only `disabled`).
- Added **breakurl**.
- **hyperref**: Fix: Several macros were made robust, `\Gauge` added.

v0.42:

Support TeX!
- Added TeX development support page, Supporting TeX development.

word-processor conversion
- Improved assitance for word-processor conversions when boolean `FormatWP` is set `true`. See section 11.

⚠ name change
  - The boolean `FormatWordProcessor` has been renamed `FormatWP`.

⚠ name change
  - The boolean `HTMLMarkFloats` has been renamed `WPMarkFloats`.

  - New booleans control whether to place additional marks around mini-pages, at the table of contents, at the LOF and LOT, and whether to print math as LaTeX source for copy/paste into the **LibreOffice Writer TeXMaths** extension.
  - Improved formatting for numerous objects. See section 11.

tabbing
- Add: `tabbing` environment.

overpic
- Add: **overpic** package. See section 300.

math
- Fix: Text copy/paste of $\mathcal{AMS}$ math environment numbers and names.
- Improved `\ensuremath`.
- MathJax with **siunitx**: Updated script and documentation.

symbols
- **textcomp**: Improved `\interrobangdown`.
- **realscripts**: Fix for subscripts in a `lateximage`.

load order
- **morewrites**: Enforces loading before **lwarp**.

v0.41:

`tabular`
- Added tabular vertical rules, subject to some limitations. See the rules section of section 9.9.
- Improved **booktabs**: Width and trim are honored.

⚠ new syntax
- Added `\mcolrowcell` for empty cells inside a `\multicolumnrow`. Use `\mcolrowcell` instead of `\mrowcell` for two-dimensional cells created by `\multicolumnrow`. Continue to use `\mrowcell` for empty cells in a `\multirow`. See section 285.2 on page 792.
- Fix: Unfinished tabular rows are automatically filled.
- Fix for tabular column specifiers while using **babel-french**. (`\NoAutoSpacing` is activated then nullified inside the tabular, due to a conflict with the tabular column parsing code.)

v0.40:

graphics, **graphicx**
- **graphics** and **graphicx** have been moved from the **lwarp** core, and are only loaded if requested with `\usepackage`.

`\includegraphics` path
- Improved **graphics** `\graphicspath` support. Multiple image directories may now be used. Refer to `.pdf` files without a file extension to allow the HTML version to use a `.svg`, `.png`, `.jpg`, or `.gif` version instead. See section 9.7.

⚠ image file extensions
- **grffile** is now directly supported instead of emulated.

bigdelim
- Fix for **bigdelim**, and improved documentation. See section 128.

symbols
- Improved LaTeX and **textcomp** symbols.

fixes
- Fix for LaTeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for **xltxtra** with XeLaTeX.

- Fixes for **tocbibind** with \simplechapter, etc.
- Fixes for \multicolumnrow and \nullfonts with older versions of **multirow** and **xparse**.
- Added \underline.

margins
- Added **adjmulticol**.

columns
- Added **cuted**, **midfloat**.

footnotes
- Added **pfnote**, **fnpos**, **dblfnote**.

tabular
- Added **stabular**, **tabls**.

sectioning
- Added **sectsty**, **anonchap**, **quotchap**.

v0.39:

title pages
- Improved the titlepage HTML code, \thanks notes, and \maketitle. **titling** is no longer required, but is still supported. The \published and \subtitle fields are no longer provided, but \AddSubtitlePublished replicates them using **titling**. See section 62.8. **authblk** is added, and should be loaded before **titling**. See section 62.

⚠ \published and \subtitle

⚠ load order

tabular
- \multirow now supports the new optional vpos argument.

multi column/row cell
- Added \multicolumnrow for combined \multicolumn and \multirow. See section 285.2.
- Tabular special cases:
  - Added \TabularMacro to mark custom macros inside tabular data cells, avoiding row corruption. See section 9.9.
  - Added \ResumeTabular for use when a tabular environment is defined inside another environment. See section 9.9.

⚠ macros inside tabular

⚠ tabular defined inside another environment

tabular
- Added **supertabular**, **xtab**, **bigstrut**, **bigdelim**.

margins
- Added **fullwidth**.

page layout
- Added **addlines**, **anysize**, **a4**, **a4wide**, **a5comb**, **textarea**, **zwpagelayout**, **typearea**, **ebook**.

v0.38:

forced single-pass compile
- Added ***lwarpmk print1*** and ***lwarpmk html1*** actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.

starred sections
- Added \ForceHTMLPage and \ForceHTMLTOC to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 9.5.2.

updated tutorial
- Modified the tutorial to use the new \ForceHTMLPage and \ForceHTMLTOC macros.

packages
- Added **appendix**, **tocbibind**, **fncychap**, **fix2col**.

font size
- Added **relsize**, **scalefnt**.

- Added **realscripts**, **metalogo**, **xltxtra**.
- Added **grffile**, **romanbar**.

page numbering
- Added **arabicfront**, **chappg**, **nonumonpart**, **nopageno**, **romanbarpagenumber**.

front & back matter
- Docs: Improved description of the use of front/back matter. See section 9.5.
- Fix: **color** requests **xcolor**.
- Fix: `\part` for **article** class.

v0.37:

`\include` for HTML
- `\include` now maintains independent `.aux` files for HTML versions.

latexmk
- **comment**, used by **lwarp**, now maintains independent cut files for print and HTML versions, helping **latexmk** to better know whether to recompile.

accents and symbols
- Improved support for LaTeX accents, **textcomp**, **siunitx** symbols.

babel-french
- Improved **babel-french** handling for load order and ~ tilde.

v0.36:

- Recorganized the documentation section regarding special cases and limitations. (Section 9)
- Improved source formatting.

boxes and frames
- `\fbox` and related now use `\fboxsep` and `\fboxrule`.
- `\makebox` and `\framebox` now use width and position.
- `\fcolorbox` and related now work inside a `lateximage`.

babel-french
- **babel-french**: Improvements for French variants, load order, footnotes, ellipses.

footnotes
- Improved footnote numbering. `lateximage` footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with MathJax.

siunitx
- Improved **siunitx** units.
- Fix for filenames while using MathJax.
- Fix for `\rule` when **xcolor** is not loaded.
- Added **transparent**, **upref**.

v0.35: Fix: `\textbf` and related.

v0.34:

⚠ Optional arguments
- `BlockClass`'s optional argument has been moved in front of the mandatory argument:

  `BlockClass[style]{class}` (NEW)

  instead of:

  `BlockClass{class}[style]` (OLD)

  This change makes it more consistent with LaTeX standards, and avoids problems with space between arguments.

⚠ Optional arguments

- Likewise, \InlineClass's optional argument now comes before the mandatory arguments:

  \InlineClass[style]{class}{text}

spans with minipages

framing minipages

- Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an \fbox containing a minipage, although with minimal HTML fomatting. See section 9.3.3. \fboxBlock is added to frame minipages, tables, and lists with full HTML formatting but no longer inline, and behaves as \fbox for print output. The fminipage environment is added for framed minipages, as an environment with full HTML formatting, and draws a framed minipage in print output. See section 9.3.5. \fbox and minipages now often work in SVG math and lateximages. MATHJAX supports \fbox, but not \fboxBlock nor fminipage.

lateximage, SVG math, tabular

- Improved compatibility between lateximage and minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox, tabular, **booktabs**.
- Improved font control for lateximagees and SVG math.

eqnarray

- Added the eqnarray environments.

verbatim packages

- **fancyvrb** is no longer required (preloaded), but is still supported.
- Added **verbatim** and **moreverb**.

framing packages

- Added **fancybox**, **boxedminipage2e** and **shadow**.

list packages

- **enumitem** is no longer required, but is still supported.
- Added **enumerate** and **paralist**.
- **titleps** is no longer required, but is still supported.
- Added **crop**.
- Added **rotfloat**, **marginfit**, and several minor packages; see the change log.

babel-french

- Adds fixed-width HTML spaces around punctuation when using **babel-french**. LuaTEX does not yet use the extra punctuation spacing.

v0.33:

- Tabular @ and ! columns now havetheir own HTML columns.
- & catcode changes are localized, perhaps causing errors about the tab alignment character &, so any definitions of macros or environments which themselves contain tabular and & must be enclosed within \StartDefiningTabulars and \StopDefiningTabulars (previously called \EndDefiningTabulars. See section 40. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the & character inside the definition. This may include the use inside conditional expressions.

- Several math environments were incorrectly placed inline. Also, for **amsmath** with SVG math, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.

v0.32: Bug fixes; no source changes needed:

- **lwarpmk** has been adjusted to work with the latest **luatex**.
- Spaces in the `\usepackage` and `\RequirePackage` package lists are now accepted and ignored.
- Fix for the **glossaries** package and `\glo@name`.

v0.31: Bug fix; no source changes needed:

- Improved compatibility with **keyfloat**, including the new `keywrap` environment.

v0.30:

⚠ **lwarp-newproject**

- **lwarp-newproject** has been removed, and its functions have been combined with **lwarp**.

  To modify existing documents, remove from the document source:

  ```
  \usepackage{lwarp-newproject}
  ```

  The **lwarp** package now produces the configuration files during print output, and also accepts the option `lwarpmk` if desired.

⚠ HTML setup changes.

- A number of macros related to HTML settings have been converted to options, and other macros and options have been renamed to create a consistent syntax:

Old Macro	New Package Option
\HomeHTMLFileName	HomeHTMLFilename
\HTMLFileName	HTMLFilename
\useLatexmk	latexmk
\warpOSwindows	OSWindows

Old Package Option	New Package Option
lwarpmklang	xindyLanguage
(new)	xindyStyle

Old Macro	New Macro
\MetaLanguage	\HTMLLanguage
\HTMLauthor	\HTMLAuthor
\NewHTMLdescription	\HTMLDescription
\SetFirstPageTop	\HTMLFirstPageTop
\SetPageTop	\HTMLPageTop
\SetPageBottom	\HTMLPageBottom
\NewCSS	\CSSFilename

- Per the above changes, in existing documents, modify the package load of **lwarp**, such as:

```
\usepackage[
    HomeHTMLFilename=index,
    HTMLFilename={},
    xindyLanguage=english
]{lwarp}
```

- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
  1. Make the changes shown above.
  2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
  3. The old file `lwarp_html.xdy` may be deleted.
- The new **lwarp** package option `xindyStyle` may be used to tell **lwarpmk** to use a custom `.xdy` file instead of `lwarp.xdy`. See section 9.5.18.
- Improvements in index processing:
  - **xindy**'s language is now used for index processing as well as glossary.
  - Print mode without **latexmk** now uses **xindy** instead of **makeindex**.
  - **texindy**/**xindy** usage depends on **pdflatex** vs **xelatex**, **lualatex**.
  - For **pdflatex** and **texindy**, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

v0.29:

- Add: `lwarpmklang` option for **lwarp-newproject** and **lwarp**. Sets the language to use while processing the glossary. (As of v0.30, this has been changed to the `IndexLanguage` option.) (As of v0.54, this has been changed to the `xindyLanguage` option.)
- Fix: `\includegraphics` when no optional arguments.

v0.28:

- `\HTMLAuthor {`⟨*name*⟩`}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Booean `HTMLDebugComments` controls whether HTML comments are added for closing `<div>`s, opening and closing sections, etc.
- Boolean `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
- Boolean `FormatWordProcessor` changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Name changed to `FormatWP` as of v0.42.
- Boolean `HTMLMarkFloats` adds text marks around floats only if the boolean `FormatWordProcessor` is `true`. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames. Name changed to `WPMarkFloats` as of v0.42.
- Updated for the new MathJax CDN repository.
- Adds **tabulary**.
- Supports the options syntax for **graphics**.
- Improved index references, now pointing exactly to their target.
- Adds **glossaries**. **lwarpmk** is modified to add `printglosssary` and `htmlglossary` actions.

3 Introduction

The **lwarp** project aims to allow a rich LaTeX document to be converted to a reasonable HTML5 interpretation, with only minor intervention on the user's part. No attempt has been made to force LaTeX to provide for every HTML-related possibility, and HTML cannot exactly render every possible LaTeX concept. Where compromise is necessary, it is desirable to allow the print output to remain typographically rich, and compromise only in the HTML conversion.

Several "modern" features of HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of JAVASCRIPT. Limited testing on older browsers shows that these new features degrade gracefully.

---

lwarp is a native LaTeX package, and operates by either patching or emulating various functions. Source-level compatibility is a major goal, but occasional user intervention is required in certain cases.

As a package running directly in LaTeX, **lwarp** has some advantages over other methods of HTML conversion. TeX itself is still used, allowing a wider range of TeX trickery to be understood. Lua expressions are still available with LuaTeX. Entire categories of LaTeX packages work as-is when used with **lwarp**: definitions, file handling, utilities, internal data structures and calculations, specialized math-mode typesetting for various fields of science and engineering, and anything generating plain-text output. Blocks of PDF output may be automatically converted to SVG images while using the same font and spacing as the original print document, directly supporting Ti*k*z and `picture`. Numerous packages are easily adapted for HTML versions, either by loading and patching the originals, or by creating nullified or emulated replacements, and all without resorting to external programming. As a result, several hundred packages have already been adapted (table 1), and an uncounted number more work as-is.

Packages have been selected according to several criteria: perceived importance, popularity lists, recent CTAN updates, CTAN topics, mention in other packages, support by other HTML conversion methods, and from sample documents taken from public archives. These include some "obsolete" packages as well.[1]

Assistance is also provided for modifying the HTML output to suit the creation of EPUB documents, and for modifying the HTML output to ease import into a word processor.

---

[1] An amazing number of decades-old packages are still in use today.

pdflatex, **xelatex**, or **lualatex** may be used, allowing **lwarp** to process the usual image formats. While generating HTML output, SVG files are used in place of PDF. Other formats such as PNG and JPG are used as-is.

SVG images may be used for math, and are also used for `picture`, Ti*k*z, and similar environments. The SVG format has better browser and e-book support than MathML (as of this writing), while still allowing for high-quality display and printing of images (again, subject to potentially bug-ridden[2] browser support).

Furthermore, SVG images allow math to be presented with the same precise formatting as in the print version. Math is accompanied by `<alt>` tags holding the LaTeX source for the expression, allowing it to be copy/pasted into other documents.[3] Custom LaTeX macros may be used as-is in math expressions, since the math is evaluated entirely inside LaTeX. An MD5 hash is used to combine multiple instances of the same inline math expression into a single image file, which then needs to be converted to SVG only a single time.

The MathJax JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MathJax allows an HTML page to display math without relying on a large number of external image files.[4] **lwarp** maintains LaTeX control for cross-referencing and equation numbering, and attempts to force MathJax to tag equations accordingly.

A **texlua** program called **lwarpmk** is used to process either the print or HTML version of the document. A few external utility programs are used to finish the conversion from a LaTeX-generated PDF file which happens to have HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

lwarp automatically generates the extra files necessary for the HTML conversion, such as CSS and `.xdy` files, and configuration files for the utility **lwarpmk**. Also included is a parallel version of the user's source document, `<sourcename>-html.tex`, which selects HTML output and then inputs the user's own source. This process allows both the printed and HTML versions to co-exist side-by-side, each with their own auxiliary files.

When requesting packages during HTML conversion, **lwarp** first looks to see if it has its own modified version to use instead of the standard LaTeX version. These `lwarp-packagename.sty` files contain code used to emulate or replace functions for HTML output.

---

[2] Firefox has had an on-again/off-again bug for quite some time regareding printing SVGs at high resolution.

[3] There seems to be some debate as to whether MathML is actually an improvement over LaTeX for sharing math. The author has no particular opinion on the matter, except to say that in this case LaTeX is much easier to implement!

[4] One SVG image file per math expression, except that duplicate inline math expressions are combined into a single file according to the MD5 hash function of its contents. A common scientific paper can easily include several thousand files, and in one case the MD5 hash cut the number of files in half and the rendering time by 30%.

3.1 Supported packages and features

Table 1 lists some of the various LaTeX features and packages which may be used. Many are tested to work as-is, some are patches for the original packages, and some are emulations written for source-level compatibility. Many are nullified as being irrelevent to HTML output.

Table 1: LaTeX–HTML generation — **lwarp** package — Supported features

Category	Status and supported features.
Engines:	DVI LaTeX, pdfLaTeX, XeLaTeX, LuaLaTeX
Classes:	**article**, **book**, **report**, **scrartcl**, **scrbook**, **scrreprt**, **memoir**.
Koma-script:	**scrextend**, **scrhack**, **scrlayer**. Others as listed below.
Memoir:	**memhfixc**
Page layout:	**2up**, **a4**, **a4wide**, **a5comb**, **addlines**, **anysize**, **atbegshi**, **blowup**, **booklet**, **bophook**, **bxpapersize**, **canoniclayout**, **clrdblpg**, **continue**, **draftcopy**, **draftfigure**, **draftwatermark**, **ebook**, **everyshi**, **fancyhdr**, **fancytabs**, **fullminipage**, **fullpage**, **fwlw**, **geometry**, **grid**, **grid-system**, **gridset**, **layaureo**, **leading**, **ltxgrid**, **nccfancyhdr**, **pagegrid**, **pagesel**, **preview**, **scrlayer-scrpage**, **textarea**, **thumb**, **thumbs**, **titleps**, **tocenter**, **turnthepage**, **twoup-gen**, **typearea**, **vmargin**, **watermark**, **zwpagelayout**. Tested to work as-is: **underlin**.
Sectioning:	Adds `FileDepth` for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation are removed from the file-names. **anonchap**, **fncychap**, **quotchap**, **section**, **sectionbreak**, **sectsty**, **titlesec**.
Table of contents, figures, tables:	Supported, with hyperlinks. **multitoc**, **shorttoc**, **titletoc**, **tocbasic**, **tocbibind**, **tocloft**, **tocstyle**.
Title page:	`\maketitle`, `titlepage`, **authblk**, **titling**.
Front & back matter:	**abstract**, **appendix**.

lwarp Supported Functions — continued

Category	Status
Indexing:	**makeindex** and **xindy** are supported, with hyperlinks. **idxlayout**, **imakeidx**, **index**, **makeidx**, **repeatindex**, **splitidx**. Tested to work as-is: **hvindex**, **sepfootnotes**.
Glossary:	**glossaries** and **xindy** are used.
Bibliography:	**babelbib**, **backref**, **biblatex**, **bibunits**, **chapterbib**, **cite**, **hypernat**, **natbib**. Tested to work as-is: **notes2bib**.
Cross-references:	**bookmark**, **breakurl**, **cleveref**, **fancyref**, **hypdestopt**, **hyperref**, **prettyref**, **titleref**, **url**, **varioref**, **xurl**.
Languages:	**babel**, **polyglossia**.
Margin notes:	**marginfit**, **marginfix**, **scrlayer-notecolumn**.
Footnotes:	Adds `FootnoteDepth` to print footnotes at section breaks. **dblfnote**, **endheads**, **endnotes**, **fnbreak**, **fnpos**, **footmisc**, **footnote**, **footnpag**, **marginnote**, **nccfoots**, **pagenote**, **parnotes**, **pfnote**, **sidenote**.
Math:	Converted to SVG images with HTML `<alt>` tags containing the LaTeX source for the math expression. MathJax supported as an alternative. **amsmath**: $\mathcal{AMS}$ environments are supported. User-defined macros are available during converson, due to native LaTeX processing.
Theorems:	Native LaTeX theorems, **amsthm**, **ntheorem**, **theorem**.
Additional math:	Math fonts via SVG images, **breqn**, **cases**, **resizegather**, **xy**. Tested to work as-is: **amscd**, **bm**, **braket**, **delarray**, **guass**, **nicematrix**, **pb-diagram**, **tikz-cd**, etc.
Display math with `\displaymathother`:	Complicated math objects in display math, such as **tikz-cd**, etc.
Units and fractions:	**nicefrac**, **SIunits**, **siunitx**, **units**, **xfrac**. Tested to work as-is: **SIunits**.

lwarp Supported Functions — continued

Category	Status
Floats:	Appear where declared. **capt-of**, **caption**, **cutwin**, **dblfloatfix**, **endfloat**, **fix2col**, **flafter**, **float**, **floatflt**, **floatrow**, **fltrace**, **hypcap**, **keyfloat**, **morefloats**, **newfloat**, **nonfloat**, **placeins**, **rotfloat**, **stfloats**, **subcaption**, **subfig**, **subfigure**, **subfloat**, **trivfloat**, **wrapfig**.
Tabular:	`tabular` environment, **array**, **arydshln**, **bigdelim**, **booktabs**, **colortbl**, **diagbox**, **longtable**, **ltablex**, **ltxtable**, **multirow**, **supertabular**, **tabularx**, **tabulary**, **threeparttable**, **xltabular**, **xtab**.
Graphics:	**graphics** and **graphicx**. `\includegraphics` supports `width`, `height`, `origin`, `angle`, and `scale` tags, and adds `class`. References to PDF files are changed to SVG, other image types are accepted as well. `\rotatebox` and `\scalebox` are supported as well as HTML can handle. **rotating** is emulated but all objects are unrotated. **picture**, **tikz**, and **xy** are converted to an SVG image. **asymptote**, **epstopdf**, **figsize**, **grffile**, **overpic**, **pdftricks**, **pst-eps**, **pstricks**. Tested to work as-is: **eepic**, **tikz-3dplot**.
xcolor:	Full package color names, any color models, and mixing. `\textcolor`, `\colorbox`, `\fcolorbox`. Enhanced for HTML compatibility.
Lists:	Standard LaTeX environments, **enumerate**, **enumitem**, **hang**, **paralist**.
Environments:	Standard LaTeX environments.
`minipage`, `\parbox`:	Some HTML5-imposed limitations. Nested minipages are supported. **pbox**.
Quotations:	**csquotes**, **epigraph**, **quoting**, **verse**.
Verbatim:	**fancyvrb**, **moreverb**, **shortvrb**, **verbatim**.
Frames:	**boxedminipage2e**, **fancybox**, **framed**, **mdframed**, **shadow**, **vertbars**.
Multi-columns:	**adjmulticol**, **multicol**, **vwcol**.

lwarp Supported Functions — continued

Category	Status
Margins:	**fullwidth**, **hanging**, **midpage**.
Line numbering:	**fnlineno**, **lineno**.
Acronyms:	**acro**, **acronym**.
Todo notes:	**changebar**, **easy-todo**, **errata**, **fixme**, **fixmetodonotes**, **todo**, **todonotes**.
Direct formatting:	`\emph`, `\textsuperscript`, `\textbf`, etc are supported. `\bfseries`, etc. are only supported in some cases. **cancel**, **ellipsis**, **hyphenat**, **lettrine**, **lips**, **lua-check-hyphen**, **luacolor**, **magaz**, **nowidow**, **pdfrender**, **realscripts**, **relsize**, **scalefnt**, **soul**, **soulpos**, **soulutf8**, **textfit**, **trimclip**, **ulem**, **underscore**, **widows-and-orphans**, **xellipsis**.
Ordinals:	**engord**, **fmtcount**, **nth**.
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for thin-unbreakable, unbreakable, `\enskip`, `\quad`, `\qquad`, `\hspace`.
Rules:	`\rule` with width, height, raise, text color.
HTML reserved characters:	`\&`, `\textless`, and `\textgreater` are converted to HTML entities.
Fonts:	Used as-is. Appear in SVG math expressions or embedded image environments.
Symbols:	Native LaTeX diacriticals, **chemgreek**, **textalpha**, **textcomp**, **textgreek**, **xunicode**. Tested to work as-is: **gensymb**.
Files:	**attachfile**, **attachfile2**, **hyperxmp**, **intopdf**, **pdfpages**, **pdfx**, **xmpincl**.

lwarp Supported Functions — continued

Category	Status
Science and engineering:	**algorithm2e**, **algorithmicx**, **axodraw2**, **bytefield**, **chemfig**, **chemformula**, **chemgreek**, **chemmacros**, **chemnum**, **listings**, **mhchem**, **phfqit**, **register**. Tested to work as-is: **blochsphere**, **bodegraph**, **bohr**, **circuitikz**, **elements**, **fast-diagram**, **hepnicenames**, **heppennames**, **linop**, **pgfgantt**, **physics**, **simpler-wick**, **slashed**, **structmech**, **tikzcodeblocks**.
Liberal arts and humanities:	**forest**, **schemata**, **vowel**, **xpiano**. Tested to work as-is: **phonrule**, **piano**, **tikz-dependency**.
Accessibility:	Nullified: **accsupp**, **axessibility**.
Working as-is:	Various utility, calculation, file, and text-only packags, such as **calc**, **fileerr**, **somedefs**, **trace**, **xspace**. Also, most math-only packages, including specialized typesetting for various fields of science and engineering.

4 Alternatives

Summarized below are several other ways to convert a LaTeX or other document to HTML. Where an existing LaTeX document is to be converted to HTML, **lwarp** may be a good choice. For new projects with a large number of documents, it may be worth investigating the alternatives before decided which path to take.

4.1 Internet class

Cls `internet` The closest to **lwarp** in design principle is the `internet` class by Andrew Stacey — an interesting project which directly produces several versions of markdown, and also HTML and EPUB. https://github.com/loopspace/latex-to-internet

4.2 TeX4ht

Prog `TeX4ht` http://tug.org/tex4ht/

Prog `htlatex`

This system uses native LaTeX processing to produce a DVI file containing special commands, and then uses additional post-processing for the HTML conversion by way of numerous configuration files. In some cases **lwarp** provides a better HTML conversion, and it supports a different set of packages. TeX4ht produces several other forms of output beyond HTML, including ODT and a direct path to EPUB.

4.3 Translators

These systems use external programs to translate a subset of LaTeX syntax into HTML. Search for each on CTAN (http:\ctan.org).

Prog `Hevea` **HEVEA:** http://hevea.inria.fr/ (not on CTAN)

Prog `TtH` **TTH:** http://hutchinson.belmont.ma.us/tth/

Prog `GELLMU` **GELLMU:** http://www.albany.edu/~hammond/gellmu/

Prog `LaTeXML` **LaTeXML:** http://dlmf.nist.gov/LaTeXML/

Prog `Plastex` **PlasTeX:** https://github.com/tiarno/plastex

Prog `LaTeX2HTML` **LaTeX2HTML:** http://www.latex2html.org/
and http://ctan.org/pkg/latex2html.

Prog `TeX2page` **TeX2page:** http://ds26gte.github.io/tex2page/index.html

Finally, GladTeX may used to directly insert LaTeX math into HTML:

Prog `GladTeX` **GladTeX:** http://humenda.github.io/GladTeX/

4.4 AsciiDoc and Asciidoctor

AsciiDoc is one of the most capable markup languages, providing enough features to produce the typical technical-writing document with cross-references, and it writes LaTeX and HTML.

Prog AsciiDoc **Asciidoctor:** http://asciidoctor.org/ (More active.)

Prog AsciiDoctor **AsciiDoc:** http://asciidoc.org/ (The original project.)

4.4.1 Asciidoctor-LaTeX

The Asciidoctor-LaTeX project is developing additional LaTeX-related features.

Asciidoctor-LateX:
http://www.noteshare.io/book/asciidoctor-latex-manual
Prog Asciidoctor-LaTeX https://github.com/asciidoctor/asciidoctor-latex

4.5 Pandoc

Prog Pandoc A markup system which also reads and writes LaTeX and HTML.

Pandoc: http://pandoc.org/

(Watch for improvements in cross-references to figures and tables.)

4.6 Word processors

Prog Word
Prog LibreOffice
Prog OpenOffice

It should be noted that the popular word processors have advanced through the years in their abilities to represent math with a LaTeX-ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success. See recent developments in Microsoft® **Word**® and LibreOffice™ **Writer**.

4.7 Commercial systems

Prog Adobe
Prog FrameMaker
Prog InDesign
Prog Flare
Prog Madcap

Likewise, several professional systems exist whose abilities have been advancing in the areas of typesetting, cross-referencing, and HTML generation. See Adobe® **FrameMaker**®, Adobe **InDesign**®, and Madcap **Flare**™.

4.8 Comparisons

AsciiDoc, Pandoc, and various other markup languages typically have a syntax which tries to be natural and human-readable, but the use of advanced features tends to require many combinations of special characters, resulting in a complicated mess of syntax. By contrast, LaTeX spells things out in readable words but takes longer to type, although integrated editors exist which can provide faster entry and a graphic user interface. For those functions which are covered by the typical markup language it is arguable that LaTeX is comparably easy to learn, while LaTeX provides many more

advanced features where needed, along with a large number of pre-existing packages which provide solutions to numerous common tasks.

Text-based document-markup systems share some of the advantages of LaTeX vs. a typical word processor. Documents formats are stable. The documents themselves are portable, work well with revision control, do not crash or become corrupted, and are easily generated under program control. Formatting commands are visible, cross-referencing is automatic, and editing is responsive. Search/replace with regular expressions provides a powerful tool for the manipulation of both document contents and structure. Markup systems and some commercial systems allow printed output through a LaTeX back end, yielding high-quality results especially when the LaTeX template is adjusted, but they lose the ability to use LaTeX macros and other LaTeX source-document features.

The effort required to customize the output of each markup system varies. For print output, LaTeX configuration files are usually used. For HTML output, a CSS file will be available, but additional configuration may require editing some form of control file with a different syntax, such as XML. In the case of **lwarp**, CSS is used, and much HTML output is adjusted through the usual LaTeX optional macro parameters, but further customization may require patching LaTeX code.

The popular word processors and professional document systems each has a large base of after-market support including pre-designed styles and templates, and often include content-management systems for topic reuse.

5 Installation

Table 2 shows the tools which are used for the LaTeX to HTML conversion. In most cases, these will be available via the standard package-installation tools.

Detailed installation instructions follow.

Table 2: Required software programs

Provided by your LaTeX distribution:

From TeXLive: http://tug.org/texlive/.

LaTeX: **pdflatex**, **xelatex**, or **lualatex**.

The lwarp package: This package.

The lwarpmk utility: Provided along with this package. This should be an operating-system executable in the same way that **pdflatex** or **latexmk** is. It is possible to have the **lwarp** package generate a local copy of **lwarpmk** called `lwarpmk.lua`. See table 3.

luatex: Used by the **lwarpmk** program to simplify and automate document generation.

xindy: The **xindy** program is used by **lwarp** to create indexes. On a MiKTeX system this may have to be acquired separately, but it is part of the regular installer as of mid 2015.

latexmk: Optionally used by **lwarpmk** to compile LaTeX code. On a MiKTeX system, **Perl** may need to be installed first.

pdfcrop: Used to pull images out of the LaTeX PDF.

Poppler PDF utilities:

pdftotext: Used to convert PDF to text.

pdfseparate: Used to pull images out of the LaTeX PDF.

pdftocairo: Used to convert images to SVG.

These might be provided by your operating-system package manager.

From Poppler: poppler.freedesktop.org.

For MacOS®, see https://brew.sh/, install **Homebrew**, then

Enter ⇒ `brew install poppler`

For Windows, see:
https://sourceforge.net/projects/poppler-win32/ and:
http://blog.alivate.com.au/poppler-windows/

Perl:

This may be provided by your operating-system package manager, and is required for some of the Poppler PDF utilities.

strawberryperl.com (recommended), perl.org

Automatically downloaded from the internet as required:

MathJax: Optionally used to display math. From: mathjax.org

5.1 Installing the lwarp package

There are several ways to install **lwarp**. These are listed here with the preferred methods listed first:

Pre-installed: Try entering into a command line:

Enter ⇒ `kpsewhich lwarp.sty`

If a path to `lwarp.sty` is shown, then **lwarp** is already installed and you may skip to the next section.

TEX Live: If using a TEX Live distribution, try installing via **tlmgr**:

Enter ⇒ `tlmgr install lwarp`

MiKTEX: If using MiKTEX:

1. To install **lwarp** the first time, use the **MikTeX Package Manager (Admin)**.
2. To update **lwarp**, use **MikTeX Update (Admin)**.
3. Either way, also update the package **miktex-misc**, which will install and update the **lwarpmk** executable.

Operating-system package: The operating-system package manager may already have **lwarp**, perhaps as part of a set of TEX-related packages.

CTAN TDS archive: **lwarp** may be downloaded from the Comprehensive TEX Archive:

1. See `http://ctan.org/pkg/lwarp` for the **lwarp** package.
2. Download the TDS archive: `lwarp.tds.zip`
3. Find the TEX `local` directory:

   TEX Live:

   Enter ⇒ `kpsewhich -var-value TEXMFLOCAL`

   MiKTEX:

   In the "Settings" window, "Roots" tab, look for a local TDS root.

   This should be something like:

   `/usr/local/texlive/texmf-local/`
4. Unpack the archive in the TDS local directory.
5. Renew the cache:

   Enter ⇒ `mktexlsr`

   — or —

   Enter ⇒ `texhash`

   Or, for WINDOWS MiKTEX, start the program called **MiKTeX Settings (Admin)** and click on the button called Refresh FNDB.

CTAN `.dtx` and `.ins` files: Another form of TEX package is `.dtx` and `.ins` source files. These files are used to create the documentation and `.sty` files.

1. See http://ctan.org/pkg/lwarp for the **lwarp** package.
2. Download the zip archive `lwarp.zip` into your own `lwarp` directory.
3. Unpack `lwarp.zip`.
4. Locate the contents `lwarp.dtx` and `lwarp.ins`
5. Create the `.sty` files:
   Enter ⇒ `pdflatex lwarp.ins`
6. Create the documentation:

```
pdflatex lwarp.dtx  (several times)
makeindex -s gglo.ist -o lwarp.gls lwarp.glo
makeindex -s gind.ist lwarp.idx
pdflatex lwarp.dtx  (several times)
```

7. Copy the `.sty` files somewhere such as the TEX Live local tree found in the previous CTAN TDS section, under the subdirectory:
   `<texlocal>/tex/latex/local/lwarp`
8. Copy `lwarp_baseline_marker.png` and `lwarp_baseline_marker.eps` to the same place as the `.sty` files.
9. Copy the documentation `lwarp.pdf` to a `source` directory in the local tree, such as:
   `<texlocal>/doc/local/lwarp`
10. Renew the cache:
    Enter ⇒ `mktexlsr`
    — or —
    Enter ⇒ `texhash`
    Or, for WINDOWS MiKTEX, start the program called **MiKTeX Settings (Admin)** and click on the button called Refresh FNDB.
11. See section 5.2.1 to generate your local copy of **lwarpmk**.
12. Once the local version of `lwarpmk.lua` is installed, it may be made available system-wide as per section 5.2.

Project-local CTAN `.dtx` and `.ins` files: The `.dtx` and `.ins` files may be downloaded to a project directory, then compiled right there, alongside the document source files. The resultant `*.sty` and `lwarpmk.lua` files may be used as-is, so long as they are in the same directory as the document source. The files `lwarp_baseline_marker.png` and `lwarp_baseline_marker.eps` must also be copied as well. This approach is especially useful if you would like to temporarily test **lwarp** before deciding whether to permanently install it.

Just testing!

5.2 Installing the lwarpmk utility

(Note: If **lwarpmk** is not already installed, it is easiest to use a local copy instead of installing it system-wide. See section 5.2.1.)

After the **lwarp** package is installed, you may need to setup the **lwarpmk** utility:

1. At a command line, try executing **lwarpmk**. If the **lwarpmk** help message appears, then **lwarpmk** is already set up. If not, it is easiest to generate and use a local copy. See section 5.2.1.
2. For MiKTEX, try updating the **miktex-misc** package. This may install the **lwarpmk** executable for you.

   Otherwise, continue with the following:
3. Locate the file `lwarpmk.lua`, which should be in the `scripts` directory of the TDS tree. On a TEX Live or MiKTEX system you may use

   Enter ⇒ `kpsewhich lwarpmk.lua`

   (If the file is not found, you may also generate a local copy and use it instead. See section 5.2.1.)
4. Create **lwarpmk**:

   Unix: Create a symbolic link and make it executable:

   (a) Locate the TEX Live binaries:
   Enter ⇒ `kpsewhich -var-value TEXMFROOT`
   This will be something like:
   `/usr/local/texlive/<year>`
   The binaries are then located in the `bin/<arch>` directory under the root:
   `/usr/local/texlive/<year>/bin/<architecture>/`
   In this directory you will find programs such as **pdflatex** and **makeindex**.

   (b) In the binaries directory, create a new symbolic link from the binaries directory to `lwarpmk.lua`:
   Enter ⇒ `ln -s <pathtolwarpmk.lua> lwarpmk`

   (c) Make the link executable:
   Enter ⇒ `chmod 0755 lwarpmk`

   WINDOWS TEX Live: Create a new `lwarpmk.exe` file:

   (a) Locate the TEX Live binaries as shown above for Unix.

   (b) In the binaries directory, make a *copy* of `runscript.exe` and call it `lwarpmk.exe` This will call the copy of `lwarpmk.lua` which is in the `scripts` directory of the distribution.

   WINDOWS MiKTEX: Create a new `lwarpmk.bat` file:

(a) Locate the binaries. These will be in a directory such as:
`C:\Program Files\MiKTeX 2.9\miktex\bin\x64`
In this directory you will find programs such as `pdflatex.exe` and `makeindex.exe`.

(b) Create a new file named `lwarpmk.bat` containing:
```
texlua "C:\Program Files\MiKTeX 2.9\scripts\lwarp\lwarp.texlua" %*
```
This will call the copy of `lwarpmk.lua` which is in the `scripts` directory of the distribution.

5.2.1 Using a local copy of lwarpmk

It is also possible to use a local version of **lwarpmk**:

1. When compiling the tutorial in section 6, use the `lwarpmk` option for the **lwarp** package:
   ```
   \usepackage[lwarpmk]{lwarp}
   ```
2. When the tutorial is compiled with **pdflatex**, the file `lwarpmk.lua` will be generated along with the other configuration files.
3. `lwarpmk.lua` may be used for this project:

   Unix:

   (a) Make `lwarpmk.lua` executable:
   Enter ⇒ `chmod 0755 lwarpmk.lua`

   (b) Compile documents with
   Enter ⇒ `./lwarpmk.lua html`
   Enter ⇒ `./lwarpmk.lua print`
   etc.

   (c) It may be useful to rename or link to a version without the `.lua` suffix.

   WINDOWS:

   Compile documents with either of the following, depending on which command shell is being used:
   Enter ⇒ `texlua lwarpmk.lua html`
   Enter ⇒ `texlua lwarpmk.lua print`
   etc.

   Or:
   Enter ⇒ `lwarpmk html`
   Enter ⇒ `lwarpmk print`
   etc.

5.3 Installing additional utilities

To test for the existence of the additional utilities:

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 2 on page 78.

Enter ⇒ `luatex --version`

Enter ⇒ `xindy --version`

Enter ⇒ `latexmk --version`

Enter ⇒ `perl --version`

Enter ⇒ `pdfcrop --version`

Enter ⇒ `pdftotext -v`

Enter ⇒ `pdfseparate --version`

Enter ⇒ `pdftocairo -v`

To install xindy, latexmk, and pdfcrop:

The TEX utilities **xindy**, **latexmk**, and **pdfcrop** may be installed in **TexLive** with **tlmgr**, installed by **MikTeX**, provided by your operating system's package manager, or downloaded from the **CTAN** archive:

http://ctan.org/pkg/xindy
http://ctan.org/pkg/latexmk
http://ctan.org/pkg/pdfcrop

Prog `pdftotext`
Prog `pdfseparate`
Prog `pdftocairo`

To install the Poppler utilities to a Unix/Linux system:

The tools from the Poppler project should be provided by your operating system's package manager.

To install the Poppler utilities to a MacOS machine:

1. Install **Homebrew** from https://brew.sh/:
   Enter ⇒
```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```
2. Install the Poppler utilities:
   Enter ⇒ `brew install poppler`

To install the Poppler utilities to a Windows machine:

1. See table 2 on page 78.
2. Download and extract the Poppler utilities **pdftotext**, **pdfseparate**, and **pdfseparate** to a directory, such as `Poppler`.
3. In the Start window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose "Edit the system environment variables" in the control panel.

5. Choose the "Environment Variables" button.
6. Choose the "Path" variable, then the "Edit" button.
7. Choose the "New" button to make an additional entry.
8. Enter the `bin` directory of the POPPLER utilities, such as:
   `C:\Users\<myname>\Desktop\Poppler\poppler-0.5_x86\poppler-0.5\bin`
   Be sure to include `\bin`.
9. Click "Ok" when done.

Prog `perl` **To install Perl to a WINDOWS machine:**

1. Download and install a version of **Perl**, such as STRAWEBERRY PERL, to a directory without a space in its name, such as `C:\Strawberry`.
2. Edit the Path as seen above for the POPPLER utilities.
3. Enter the `bin` directory of the **Perl** utility, such as:
   `C:\Strawberry\perl\bin`
   Be sure to include `\bin`.
4. Click "Ok" when done.

Any utilities installed by hand must be added to the PATH.

6 Tutorial

This section shows an example of how to create an **lwarp** document.

Need help?

The index to this document contains several hundred custom entries. Also included are automated entries for each package, macro, environment, counter, boolean, and other objects; individually and also sorted by category. A Troubleshooting section is also available.

6.1 Starting a new project

1. Create a new project directory called `tutorial`.

File `tutorial.tex`

2. Inside the `tutorial` directory, create a new file called `tutorial.tex`. This may be done several ways:

   Copy from the documentation PDF:

   A listing is in fig. 1, which may be copied/pasted from the figure directly into your own editor, depending on the quality of the PDF viewer and editor, or:

   Copy from the lwarp documentation directory:

File `lwarp_tutorial.txt`

⚠ Note: `.txt` suffix!

   Another copy may be found by entering into a command line:

   Enter ⇒ `texdoc -l lwarp_tutorial.txt`

   This should be in the `doc/latex/lwarp/` directory along with this PDF documentation. Copy `lwarp_tutorial.txt` directly into your `tutorial` directory, renamed as `tutorial.tex`.

⚠ Bad formatting!

   When using WINDOWS, use an editor other than Notepad, since Notepad does not accept the end-of-line from a Unix text file.

3. Compile the project:

   Enter ⇒ `pdflatex tutorial.tex`

   (several times)

   (**xelatex** or **lualatex** may be used as well. **lwarp** also supports DVI **latex** for use with `.eps` images.)

4. View the resulting `tutorial.pdf` with a PDF viewer.

A number of new files are created when `tutorial.tex` is compiled, as shown in table 3. These files are created by the `lwarp` package.

(Two of the new files are configuration files for the helper program **lwarpmk**. Whenever a print version of the document is created, the configuration files for **lwarpmk** are updated to record the operating system, LaTeX engine (**latex**, **pdflatex**, **xelatex**, or **lualatex**), the filenames of the source code and HTML output, and whether the additional helper program **latexmk** will be used to compile the document.)

Figure 1: tutorial.tex listing

Note: There are two pages!

```
% Save this as tutorial.tex for the lwarp package tutorial.

\documentclass{book}

\usepackage{iftex}

% --- LOAD FONT SELECTION AND ENCODING BEFORE LOADING LWARP ---

\ifPDFTeX
\usepackage{lmodern}            % pdflatex
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\else
\usepackage{fontspec}           % XeLaTeX or LuaLaTeX
\fi

% --- LWARP IS LOADED NEXT ---
\usepackage[
%   HomeHTMLFilename=index,     % Filename of the homepage.
%   HTMLFilename={node-},       % Filename prefix of other pages.
%   IndexLanguage=english,      % Language for xindy index, glossary.
%   latexmk,                    % Use latexmk to compile.
%   OSWindows,                  % Force Windows. (Usually automatic.)
%   mathjax,                    % Use MathJax to display math.
]{lwarp}
% \boolfalse{FileSectionNames}  % If false, numbers the files.

% --- LOAD PDFLATEX MATH FONTS HERE ---

% --- OTHER PACKAGES ARE LOADED AFTER LWARP ---
\usepackage{makeidx} \makeindex
\usepackage{xcolor}             % (Demonstration purposes only.)
\usepackage{hyperref,cleveref}  % LOAD THESE LAST!

% --- LATEX AND HTML CUSTOMIZATION ---
\title{The Lwarp Tutorial}
\author{Some Author}
\setcounter{tocdepth}{2}        % Include subsections in the \TOC.
\setcounter{secnumdepth}{2}     % Number down to subsections.
\setcounter{FileDepth}{1}       % Split \HTML\ files at sections
\booltrue{CombineHigherDepths}  % Combine parts/chapters/sections
\setcounter{SideTOCDepth}{1}    % Include subsections in the side\TOC
\HTMLTitle{Webpage Title}       % Overrides \title for the web page.
\HTMLAuthor{Some Author}        % Sets the HTML meta author tag.
```

```
\HTMLLanguage{en-US}            % Sets the HTML meta language.
\HTMLDescription{A description.}% Sets the HTML meta description.
\HTMLFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\HTMLPageTop{\fbox{LOGO}}
\HTMLPageBottom{Contact Information and Copyright}
\CSSFilename{lwarp_sagebrush.css}

\begin{document}

\maketitle                      % Or titlepage/titlingpage environment.

% An article abstract would go here.

\tableofcontents                % MUST BE BEFORE THE FIRST SECTION BREAK!
\listoffigures

\chapter{First chapter}

\section{A section}

This is some text which is indexed.\index{Some text.}

\subsection{A subsection}

See \cref{fig:withtext}.

\begin{figure}\begin{center}
\fbox{\textcolor{blue!50!green}{Text in a figure.}}
\caption{A figure with text\label{fig:withtext}}
\end{center}\end{figure}

\section{Some math}

Inline math: $r = r_0 + vt - \frac{1}{2}at^2$
followed by display math:
\begin{equation}
a^2 + b^2 = c^2
\end{equation}


\begin{warpprint}   % For print output ...
\cleardoublepage    % ... a common method to place index entry into TOC.
\phantomsection
\addcontentsline{toc}{chapter}{\indexname}
\end{warpprint}
\ForceHTMLPage      % HTML index will be on its own page.
\ForceHTMLTOC       % HTML index will have its own toc entry.
\printindex

\end{document}
```

Table 3: Files created along with the print version

`tutorial.pdf`**:** The PDF output from LaTeX. The print version of the document.

`tutorial_html.tex`**:** A small `.tex` file used to create a parallel HTML version of the document, which co-exists with usual the PDF version, and which will have its own auxiliary files. In this way, both PDF and HTML documents may co-exist side-by-side.

Auxiliary files: The usual LaTeX files `.aux`, `.log`, `.out`, `.toc`, `.lof`, `.idx`. When an HTML version of the document is created, `_html` versions of the auxiliary files will also be generated.

`lwarpmk.conf`**:** A configuration file for **lwarpmk**, which is used to automate the compilation of PDF or HTML versions of the document.

`tutorial.lwarpmkconf`**:** Another configuration file used by **lwarpmk**, which is only useful if you wish to have several projects residing in the same directory.

`.css files`**:** `lwarp.css`, `lwarp_formal.css`, `lwarp_sagebrush.css` These files are standard for **lwarp**, and are not meant to be modified by the user.

`sample_project.css`**:** An example of a user-customized css file, which may be used for project-specific changes to the **lwarp** defaults.

`lwarp.ist`**:** Used by **lwarp** while creating an index using **makeindex**. This file should not be modified by the user. A custom file may be used instead, if necessary.

`lwarp.xdy`**:** Used by **lwarp** while creating an index using **xindy**. This file should not be modified by the user. A custom file may be used instead, if necessary.

`lwarp_one_limage.txt`**:** For WINDOWS only. Used to process SVG images in the background. Copied to `lwarp_one_limage.cmd` when images are generated.

`lwarp_mathjax.txt`**:** Inserted into the HTML files when MATHJAX is used to display math. This file should not be modified by the user.

`comment.cut`**:** A temporary file used by **lwarp** to conditionally process blocks of text. This file may be ignored.

When the `lwarpmk` option is given to the **lwarp** package:

`lwarpmk.lua`**:** A local copy of the **lwarpmk** utility.

On Unix-related operating systems this file must be made executable:

```
chmod u+x lwarpmk.lua
```

This may be useful to have to archive with a project for future use.

6.2 Compiling the print version with lwarpmk

The **lwarpmk** utility program is used to compile either the printed or the HTML version of the document.

`lwarpmk print` is used to recompile a printed version of the document.

1. Re-compile the print version:

   Enter ⇒ `lwarpmk print`

   lwarpmk prints an introduction then checks to see if the document must be recompiled. If it seems that the files are up-to-date, then **lwarpmk** informs you of that fact and then exits.

2. Make a small change in the original document, such as adding a space character.

3. Recompile again.

   Enter ⇒ `lwarpmk print`

   The document is recompiled when a change is seen in the source. Several compilations may be necessary to resolve cross-references.

4. Force a recompile to occur.

   Enter ⇒ `lwarpmk again`

   Enter ⇒ `lwarpmk print`

   `lwarpmk again` updates the date code for the file, triggering a recompile the next time the document is made.[5]

5. Process the index. [6] [7]

   Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

   Enter ⇒ `lwarpmk print`

7. To force a single recompile when needed, even if no changes were detected:

   Enter ⇒ `lwarpmk print1`

Note that the HTML customization commands are ignored while making the print version.

[5] Although, when using the utility **latexmk** (introduced later), the changed date is ignored and an actual change in contents must occur to cause a recompile.

[6] The command ***`lwarpmk printglossary`*** is also available to process a glossary produced with the glossaries package. See section 9.5.10.

[7] Also see section 9.5.12 for index options.

6.3 Compiling the HTML version with lwarpmk

`lwarpmk html` is used to recompile an HTML version of the document.

1. Compile the HTML version:

   Enter ⇒ `lwarpmk html`

   (a) **lwarpmk** uses LaTeX to process `tutorial_html.tex` to create `tutorial_html.pdf`.

   (b) **pdftotext** is then used to convert to the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.

   (c) **lwarpmk** manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`[8], `Some-math.html`, and the document's index in `_Index.html`.[9]

2. View the HTML page in a web browser.

   Open the file `tutorial.html` in a web browser.

   math

   Note that math is still displayed as its `alt` tag, which is the plain-text LaTeX source, until the images of the math expressions have been generated. Math may be displayed as SVG images or by a MathJax script, as seen in sections 6.4 and 6.5.

3. Force a recompile:

   Enter ⇒ `lwarpmk again`

   Enter ⇒ `lwarpmk html`

   Enter ⇒ `lwarpmk print`

4. Process the HTML index and recompile: [10][11]

   Enter ⇒ `lwarpmk htmlindex`

   Enter ⇒ `lwarpmk html`

   `_Index.html` is updated for the new LaTeX index.

5. Reload the web page to see the added index.

6. To force a single recompile when needed, even if no changes were detected:

   Enter ⇒ `lwarpmk html1`

[8] `First-chapter.html` also contains the first section, even though the second section is its own HTML page. This behavior is controlled by the boolean `CombineHigherDepths`.

[9] `index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

[10] The command ***`lwarpmk htmlglossary`*** is also available to process a glossary produced with the glossaries package. See section 9.5.10.

[11] Also see section 9.5.12 for index options.

6.4 Generating the SVG images

math as SVG images

By default **lwarp** represents math as SVG images with the LaTeX source included in `alt` attributes. In this way, the math is displayed as it was drawn by LaTeX, and the LaTeX source may be copied and pasted into other documents.

`picture` and Ti*k*z

lwarp uses the same mechanism for `picture` and Ti*k*z environments.

1. Create the SVG images:

   Enter ⇒ `lwarpmk limages`

   Enter ⇒ `lwarpmk html`

2. Move to the tutorial's HTML math page and reload the document in the browser.
3. The math images are displayed using the same font and formatting as the printed version.
4. Copy/paste a math expression into a text editor to see the LaTeX source.

⚠ adding/removing

When a math expression, `picture`, or Ti*k*z environment is added or removed, the SVG images must be re-created by entering ***`lwarpmk limages`*** to maintain the proper image-file associations. Inline SVG math may be hashed and thus not need to be recreated, but display math and objects such as Ti*k*z may move to new image numbers when the document is changed.

Before attempting to create the SVG image files, **lwarpmk** verifies that the HTML version of the document exists and has correct internal image references.[12] If it is necessary to recompile the document's HTML version, **lwarpmk** will inform so with an error message.

⚠ HTML instead of images

If HTML appears where an SVG image should be, recompile the document one more time to get the page numbers back in sync, then remake the images one more time.

⚠ page counter

Incorrect SVG images will also occur if the document changes the `page` counter:

```
\setcounter{page}{<value>}
```

The `page` counter must *not* be adjusted by the user.

⚠ Lots of files!

Expressing math as SVG images has the advantage of representing the math exactly as LaTeX would, but has the disadvantage of requiring an individual file for each math expression. For inline math, and some other objects, **lwarp** uses an MD5 hash on its LaTeX source to combine multiple instances of identical inline expressions into a single image file, but display math and other environments such as `picture` and Ti*k*z require one image file each. For a document with a large amount of math, see section 6.5 to use MathJax instead.

[12] This becomes important when dealing with a document containing thousands of images.

6.5 Using MathJax for math

math with MathJax

Math may also be represented using the MathJax JavaScript project.

1. In the tutorial's source code, uncomment the `mathjax` package option for **lwarp**:

   ```
   mathjax, % Use MathJax to display math.
   ```

2. Recompile

   Enter ⇒ `lwarpmk html`

3. Reload the math page.

⚠ MathJax requirements

MathJax requires web access unless a local copy of MathJax is available, and it also requires that JavaScript is enabled for the web page. The math is rendered by MathJax. Right-click on math to see several options for rendering, and for copying the LaTeX source.

While using MathJax has many advantages, it may not be able to represent complex expressions or spacing adjustments as well as LaTeX, and it may not support some math-related packages.

6.6 Changing the CSS style

For a formal css style, add to the preamble:

```
\usepackage{lwarp}
...
\CSSFilename{lwarp_formal.css}
...
\begin{document}
```

For a modern css style, `lwarp_sagebrush.css` is also provided:

```
\CSSFilename{lwarp_sagebrush.css}
```

See section 8.5 for more information about modifying the CSS styling of the document.

6.7 Customizing the HTML output

A number of settings may be made to control the HTML output, including filename generation, automatic compilation, math output, document splitting, meta data, and page headers and footers.

See section 8.4 for more information.

6.8 Using latexmk

latexmk is a LaTeX utility used to monitor changes in source files and recompile as needed.

1. In the tutorial's source code uncomment the `latexmk` option for the **lwarp** package:

   ```
   latexmk, % Use latexmk to compile.
   ```

2. Recompile the printed version of the document.

   Enter ⇒ `lwarpmk print`

   lwarp updates its own configuration files (`lwarpmk.conf` and `tutorial.lwarpmkconf`) whenever the printed version of the document is compiled. These configuration files remember that **lwarpmk** should use **latexmk** to compile the document.

3. Recompile the document.

   Enter ⇒ `lwarpmk print`

   and/or

   Enter ⇒ `lwarpmk html`

Changes are detected by comparing checksums rather than modification times, so `lwarpmk again` will not trigger a recompile, but **latexmk** has a much better awareness of changes than the **lwarpmk** utility does and it is likely to correctly know when to recompile. A recompile may be forced by making a small change to the source, and a single recompile may be forced with:

forced single-pass recompile

Enter ⇒ `lwarpmk print1`

and/or

Enter ⇒ `lwarpmk html1`

6.9 Using XeLaTeX or LuaLaTeX

XeLaTeX or LuaLaTeX may be used instead of LaTeX.

1. Remove the auxiliary files for the project:

   Enter ⇒ `lwarpmk cleanall`

2. Use **xelatex** or **lualatex** to compile the printed version a single time.

   Enter ⇒ `xelatex tutorial.tex`

   -or-

   Enter ⇒ `lualatex tutorial.tex`

   When the compile occurs, the configuration files for **lwarpmk** are modified to remember which TeX engine was used. XeLaTeX or LuaLaTeX will be used for future runs of **lwarpmk**.

3. To recompile the document:

   Enter ⇒ `lwarpmk print`

   -and-

   Enter ⇒ `lwarpmk html`

4. Also remember to update the indexes and recompile again:

   Enter ⇒ `lwarpmk htmlindex`

   Enter ⇒ `lwarpmk html`

   Enter ⇒ `lwarpmk printindex`

   Enter ⇒ `lwarpmk print`

6.10 Using DVI LaTeX

Traditional DVI LaTeX may also be used along with `.eps` image files. An SVG version of each image must also be provided. **lwarpmk** may be used to convert image formats.

To convert EPS files to PDF:

Enter ⇒ `lwarpmk epstopdf *.eps` (or a list of files)

To convert PDF files to SVG:

Enter ⇒ `lwarpmk pdftosvg *.pdf` (or a list of files)

6.11 Using a glossary

lwarp supports the **glossaries** package, although this tutorial does not supply an example.

To process the glossary for the print version:

Enter ⇒ `lwarpmk printglossary`

⚠ (If ***`makeglossaries`*** is not found, see section 9.5.10.)

To process the glossary for the HTML version:

Enter ⇒ `lwarpmk htmlglossary`

In each case, the document will have to be recompiled afterwards:

Enter ⇒ `lwarpmk html1`

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk print1`

Enter ⇒ `lwarpmk print`

See section 9.5.10 to set options for processing glossaries.

6.12 Cleaning auxiliary files

To remove the auxiliary files `.aux`, `.toc`, `.lof`, `.lot`, `.idx`, `.ind`, `.log`, and `.gl*`:

Enter ⇒ `lwarpmk clean`

6.13 Cleaning auxiliary and output files

To remove the auxiliary files, and also remove the `.pdf` and `.html` files:

Enter ⇒ `lwarpmk cleanall`

6.14 Cleaning the images from the `lateximages` directory

The `lateximage` directory contains SVG images automatically generated for inline and display math, **tikz**, etc. To remove all the images from the `lateximages` directory:

Enter ⇒ `lwarpmk cleanlimages`

6.15 Converting PDF or EPS images to SVG

HTML cannot display PDF or EPS images, so any external PDF graphics images must be converted to SVG format. **pdftocairo** and **epstopdf** may be used one image at a time, but **lwarpmk** also provides a way to convert PDF or EPS images in bulk:

Enter ⇒ `lwarpmk epstopdf *.eps` (or a list of files)

Enter ⇒ `lwarpmk pdftosvg *.pdf` (or a list of files)

Be sure to always provide SVG files for HTML output.

6.16 Creating HTML from an incomplete compile

During testing it may be useful to finish the HTML conversion even when the document had errors and did not compile successfully. To attempt an HTML conversion of an incomplete document:

Enter ⇒ `lwarpmk pdftohtml [-p project]`

6.17 Processing multiple projects in the same directory

It is possible to have several projects in the same directory. **lwarpmk** has an optional parameter which is the document to compile.

To create each project:

Enter ⇒ `pdflatex project_a`

Enter ⇒ `pdflatex project_b`

Each project is given its own configuration file:

`project_a.lwarpmkconf, project_b.lwarpmkconf`

To compile each project with `lwarkmk`:

Enter ⇒ `lwarpmk print -p project_a`

Enter ⇒ `lwarpmk html -p project_b`

6.18 Using the make utility

lwarpmk has an action which may be useful for integration with the common **make** utility:

`lwarpmk pdftohtml [-p project]`

make may be used to compile the code to PDF with HTML tags (`project_html.pdf`), then **lwarpmk** may be used to convert each target to HTML files.

7 Converting an existing document

To convert an existing document for use with **lwarp**:

1. Arrange the document in the following order:
   (a) Declare the `\documentclass`.
   (b) Load text fonts.
   (c) Load **inputenc** or **inputenx**, **fontenc**, and/or **fontspec**.
   (d) Load **lwarp**.
   (e) Load remaining packages.
2. Modify the document:
   (a) Avoid the `scale` option. Change:
   ```
   \includegraphics[scale=<xx>]
   ```
   to:
   ```
   \includegraphics[width=<yy>\linewidth]
   ```
   (b) Possible changes to `tabular` environments include * columns, **multirow**, **longtable**, **supertabular**, **xtab**, **bigdelim**. See section 9.9.
   (c) Possible option clashes with **memoir**. See section 9.12.
   (d) If using indexes, see section 9.5.12.
   (e) If using many indexes, glossaries, `.aux` files, etc., see section 9.5.12 regarding **morewrites**. If **morewrites** is already used, be sure to add the setup with `allocate=10`.
   (f) Other changes as per Special cases and limitations, section 9.
3. Convert any PDF images to SVG. See section 9.7.
4. Manually compile the print version with **latex**, **pdflatex**, **lualatex**, or **xelatex**.
5. `lwarpmk print` to finish the print version.
6. `lwarpmk html` to create the HTML version.
7. `lwarpmk limages` to create the SVG images of any SVG math, `lateximage`, Ti*k*Z, etc.

Need help?

The index to this document contains several hundred custom entries. Also included are automated entries for each package, macro, environment, counter, boolean, and other objects; individually and also sorted by category. A Troubleshooting section is also available.

8 Additional details

8.1 Shell escape

Opt `-shell-escape`

Some documents require the use of an external program, which is allowed when using the `-shell-escape` command-line option. When the document is first compiled manually, and also whenever the print version is recompiled, **lwarp** detects and remembers whether shell escape is enabled. If so, it will also be enabled when the document is recompiled with **lwarpmk**.

8.2 Font and UTF-8 support

lwarp uses **pdftotext** to convert PDF output into UTF-8-encoded text. This process requires that UTF-8 information be embedded in the PDF file, which usually prevents the use of older bit-mapped fonts, and of older packages such as **ae**. The **lwarp** option `pdftotextEnc` may also be useful in some situations. See section 8.3.

pdflatex, T1, UTF8

While using **pdflatex**, **fontenc** is automatically loaded with T1 encoding. **fontenc** may be loaded with an additional encoding after **lwarp**. **inputenc** is automatically loaded with UTF8 encoding if if has not yet been loaded, but may also be specified with another encoding such as `latin1`. See the next section regarding index encoding.

vector fonts
Computer Modern
⚠ **pdflatex**

While using **pdflatex**, if no font-related package is specified, the default bit-mapped Computer Modern font is used, so simply add

```
usepackage{lmodern}
```

to the preamble to enable the related vector font instead, or use

```
\usepackage{dejavu}
```

or other font packages, which may provide an increased coverage of Unicode mappings. Avoid bit-mapped fonts.

⚠ **xelatex** and **lualatex**

XeLaTeX and LuaLaTeX users must use the **fontspec** package. Do NOT use **fontenc**!

Place **fontspec** or **fontenc**, **xunicode**, and other font and UTF-8 related commands after the `\documentclass` command and before `\usepackage{lwarp}`.

⚠ package conflicts

In some cases, a package conflict may require that a font package be loaded after **lwarp**, which should work as well:

1. `documentclass{article/book/report}` comes first, followed by any of:
2. Font and UTF-8 related commands:

- For XeLaTeX or LuaLaTeX:
  - **fontspec** and font choices

    Pkg fontspec

    ligatures

    lwarp sets the following to turn off TeX ligatures during the generation of HTML tags, and turn off common ligatures in regular text, since older browsers may not display them correctly and newer browsers can automatically re-create them.

```
\defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
```

- For **pdflatex**:
  - (a) `\usepackage{lmodern}`, or other font-related packages

    Pkg lmodern
  - (b) `\usepackage[T1]{fontenc}`

    Pkg fontenc
  - (c) `\usepackage[utf8]{inputenc}`, or `latin1`, etc. Or use **inputenx**.

    Pkg inputenc

    Pkg inputenx
  - (d) `\usepackage{newunicodechar}` along with related definitions.

    Pkg newunicodechar
  - (e) To assist with the PDF-HTML conversion:
    - i. `\input glyphtounicode.tex`

      File glyphtounicode.tex
    - ii. `\input glyphtounicode-cmr.tex% from the pdfx package`
    - iii. `\pdfgentounicode=1`
  - (f) Another option to assist with the PDF-HTML conversion:
    - `\usepackage{cmap}`

      Pkg cmap
  - (g) `\usepackage{textcomp}`

    Pkg textcomp

3. `\usepackage{lwarp}` (section 8.3) goes after any of the above, followed by:
4. `\usepackage{newtxmath}` or other math-related font packages. Many of these load **amsmath**, which must be loaded after **lwarp**, so they must also be loaded after **lwarp**.

⚠ **fontspec** with traditional font packages

5. `\setmonofont{TeX Gyre Cursor}` or similar may be required if using XeLaTeX or LuaLaTeX and **fontspec** along with traditional font packages such as **txfonts**, **newtxtext**, etc. This is required to turn off the monospaced font's ligatures with **fontspec** after loading the traditional font packages. Monospaced output ligatures must be turned off to produce the correct HTML characters.
6. ... the rest of the preamble and the main document.

8.2.1 Indexes, glossaries, and encoding

lwarp uses the **xindy** program to processes indexes. **xelatex** and **lualatex** use **xindy** and **pdflatex** uses **texindy**.

The **lwarp** option `xindyLanguage` may be used to set the language option for **xindy**, and the `xindyCodepage` option may be used to set the codepage option for **xindy**. These are used for index generation.

8.3 lwarp package loading and options

lwarp supports **book**, **report**, and **article** classes, as well as the equivalent Koma-script classes and **memoir**.

Pkg `lwarp` Load the **lwarp** package immediately after the font and UTF-8 setup commands.

Opt `mathsvg` / Opt `mathjax` **`mathsvg`, `mathjax`:** For math display, select `mathsvg` (default), or `mathjax`. For more information about the math options, see section 9.6.

Opt `latexmk` Default: `false` **`latexmk`:** Tells **lwarpmk** to use **latexmk** to recompile the document several times if necessary. Otherwise, **lwarpmk** attempts to determing for itself whether to recompile. See section 8.4.

Opt `dvips` Default: `false` **`dvips`:** Tells **lwarpmk** to use **dvips** and **ps2pdf** to convert DVI output to PDF.

Opt `dvipdfm` Default: `false` **`dvipdfm`:** Tells **lwarpmk** to use **dvipdfm** to convert DVI output to PDF.

Opt `dvipdfmx` Default: `false` **`dvipdfmx`:** Tells **lwarpmk** to use **dvipdfmx** to convert DVI output to PDF.

Opt `HomeHTMLFilename` Default: `{}` **`HomeHTMLFilename`:** See section 8.4.

Opt `HTMLFilename` Default: `{}` **`HTMLFilename`:** See section 8.4.

Opt `makeindex` Default: `makeindex` **`makeindex`:** Sets `PrintIndexCmd`, `HTMLIndexCmd`, and `LatexmkImageCmd` to use **makeindex** when generating indexes with ***`lwarpmk printindex`***, ***`lwarpmk htmlindex`***, or **latexmk**. If neither `makeindex` nor `xindy` is used, `makeindex` is assumed.

Opt `xindy` Default: `makeindex` **`xindy`:** Sets `PrintIndexCmd`, `HTMLIndexCmd`, and `LatexmkImageCmd` to use **xindy** when generating indexes with ***`lwarpmk printindex`***, ***`lwarpmk htmlindex`***, or **latexmk**.

Opt `makeindexStyle` Default: `lwarp.ist` **`makeindexStyle`:** If you wish to use a custom `.ist` file for index generation, see section 24.

Opt `xindyStyle` Default: `lwarp.xdy` **`xindyStyle`:** If you wish to use a custom `.xdy` file for index generation, see section 24.

Opt `xindyLanguage` Default: `english` **`xindyLanguage`:** If using an index or glossary, see section 24.

Opt `xindyCodepage` Default: `utf8` **`xindyCodepage`:** If using an index, see section 24.

Opt `PrintIndexCmd` Default: `<automatic>` **`PrintIndexCmd`:** Sets the shell commands executed by ***`lwarpmk printindex`***. If not specified, will be set by the selection of `makeindex` or `xindy`. May be used to specify the creation of multiple indexes. See section 9.5.12.

Table 4: **lwarp** package options

Option	Description
`mathsvg`	Show math using SVG images.
`mathjax`	Show math using MATHJAX.
`latexmk`	Use **latexmk** for compiling documents.
`dvips`	Use **dvips** and **ps2pdf** to convert DVI documents.
`dvipdfm`	Use **dvipdfm** to convert DVI documents.
`dvipdfmx`	Use **dvipdfmx** to convert DVI documents.
`HomeHTMLFilename`	The filename of the home page.
`HTMLFilename`	A prefix for the filenames of the remaining web pages.
For indexing (section 9.5.12) and glossaries (section 9.5.10):	
`makeindex`	Use **makeindex** to generate indices.
`xindy`	Use **xindy** to generate indices.
`makeindexStyle`	Set a custom style for **makeindex**.
`xindyStyle`	Set a custom style for **xindy**.
`xindyLanguage`	The **xindy** language option used for index generation.
`xindyCodepage`	The **xindy** codepage option used for index generation.
`PrintIndexCmd`	Shell commands executed by ***`lwarpmk printindex`***.
`HTMLIndexCmd`	Shell commands executed by ***`lwarpmk htmlindex`***.
`LatexmkIndexCmd`	Shell commands executed by **latexmk**.
`GlossaryCmd`	Shell command executed by ***`lwarpmk printglossary`*** and ***`lwarpmk htmlglossary`***.
Seldom necessary:	
`OSWindows`	Force compatibility with MS-WINDOWS.
`pdftotextEnc`	Set the encoding for **pdftotext**.
`lwarpmk`	Generate a local copy of `lwarpmk.lua`.
Used internally by **lwarp**:	
`warpprint`	Generate print output, and also generate configuration files.
`warpHTML`	Generate HTML output.
`BaseJobname`	The `\jobname` to use. Set to the `\jobname` of the printed version even while generating HTML.

Examples:

```
makeindex -s lwarp.ist projectname.idx                (makeindex)
xindy -M lwarp.xdy -L english -C utf8 projectname.idx      (xindy)
```

automatic setting

The use of the `makeindex` or `xindy` options sets `PrintIndexCmd` to sensible values for each of those programs while compiling a single index. **lwarp**'s `makeindexStyle`, `xindyStyle`, `xindyLanuage`, and `xindyCodepage` options will be used if specified.

⚠ xindy

If specifying `PrintIndexCmd` manually, be sure to assign an **xindy** language and codepage with the `-L` and `-C` **xindy** options, as the **lwarp** `xindyLanguage` and `xindyCodepage` options are not used for the `PrintIndexCmd` option when it is set manually.

This option is stored in the configuration files `lwarpmk.conf` and `*.lwarpmkconf`, and is then passed by the ***lwarpmk printindex*** command to the operating system to compile the print indexes. Since the command string is parsed by TEX, written to a file, read from the file by LuaTEX, and finally passed to the operating system, any attempt at quoting will be problematic. For complicated commands, it would be best to create a shell script, and simply refer to the script with the **lwarp** `PrintIndexCmd` option.

Opt `HTMLIndexCmd`
Default: `<automatic>`

`HTMLIndexCmd:` Sets the shell commands executed by ***`lwarpmk htmlindex`***. If not specified, will be set by the selection of `makeindex` or `xindy`. May be used to specify the creation of multiple indexes. See section 9.5.12.

⚠ filenames

Example settings are similar to `PrintIndexCmd`, but append `_html` to the filenames:

```
makeindex -s lwarp.ist projectname_html.idx           (makeindex)
xindy -M lwarp.xdy -L english -C utf8 projectname_html.idx (xindy)
```

automatic setting

The use of the `makeindex` or `xindy` options sets `HTMLIndexCmd` to sensible values for each of those programs while compiling a single index. **lwarp**'s `makeindexStyle`, `xindyStyle`, `xindyLanuage`, and `xindyCodepage` options will be used if specified.

⚠ xindy

If specifying `HTMLIndexCmd` manually, be sure to assign an **xindy** language and codepage with the `-L` and `-C` **xindy** options, as the **lwarp** `xindyLanguage` and `xindyCodepage` options are not used for the `HTMLIndexCmd` option when it is set manually.

As with `PrintIndexCmd`, to generate complicated indexes it may be worthwhile to use a shell script, then refer to that script with `HTMLIndexCmd`.

Opt `LatexmkIndexCmd`
Default: `<automatic>`

`LatexmkIndexCmd:` Sets the shell commands executed by **latexmk**. Unlike `PrintIndexCmd` and `HTMLIndexCmd`, `LatexmkIndexCmd` does not include any filenames, which will be provided instead by **latexmk**. See section 9.5.12.

Example settings are similar to `PrintIndexCmd`, but without a filename:

```
makeindex -s lwarp.ist                                (makeindex)
xindy -M lwarp.xdy -L english -C utf8                      (xindy)
```

automatic setting

The use of the `makeindex` or `xindy` options sets `LatexmkIndexCmd` to ei-

ther of the two settings show above. **lwarp**'s `makeindexStyle`, `xindyStyle`, `xindyLanuage`, and `xindyCodepage` options will be used if specified. Unlike `PrintIndexCmd` and `HTMLIndexCmd`, **latexmk** uses either of the single-line settings of `LatexmkIndexCmd` shown above to compile each of multiple indexes if necessary.

⚠ xindy

If specifying `LatexmkIndexCmd` manually, be sure to assign an **xindy** language and codepage with the `-L` and `-C` **xindy** options, as the **lwarp** `xindyLanguage` and `xindyCodepage` options are not used for the `LatexmkIndexCmd` option when it is set manually.

Opt `GlossaryCmd`
Default: *`makeglossaries`*

`GlossaryCmd`: Sets the shell command executed by *`lwarpmk printglossary`* and *`lwarpmk htmlglossary`*. The print or HTML glossary filename is appended to this command. See section 9.5.10.

Opt `OSWindows`

`OSWindows`: **lwarp** attempts to automatically sense WINDOWS, but it may be forced with this option. See section 8.6.

Opt `pdftotextEnc`
Default: `UTF-8`

`pdftotextEnc`: Used to specify the encoding used by **pdftotext** during the PDF-HTML conversion. In most situations, the default is the correct choice.

Opt `lwarpmk`

`lwarpmk`: If you wish to have **lwarp** generate a local copy of `lwarpmk.lua` for archival or local-installation purposes, compile the print version with the `lwarpmk` option set. See section 24.

The following options are used internally by **lwarp**, and usually are not used in the user's document:

Opt `warpprint`
Opt `warpHTML`

`warpprint` and `warpHTML`: Usually controlled by **lwarpmk**, and not set in the document. Select the `warpprint` option to generate print output (default), or the `warpHTML` option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual **pdflatex**, etc. When **lwarp** is loaded in print mode, it creates `<project>_html.tex`, which sets the `warpHTML` option before calling the user's source code `<project>.tex`. In this way, `<project>.tex` can `\usepackage{lwarp}` without any options to create a printed version, while `<project>_html.tex` will create an HTML version.

Opt `BaseJobname`
Default: `\jobname`

`BaseJobname`: Not intended for the user. Used internally by **lwarp** when creating the `*_html.tex` file used to compile the HTML version. See section 24.

8.4 Customizing the HTML output

⚠ Placement! Table 5 shows several settings may be used to customize the HTML output. Watch for the correct placement of each!

⚠ Changes! Note that if changes are made, it is best to first:

1. Clear all the HTML, PDF, and auxiliary files:
   Enter ⇒ `lwarpmk clearall`
2. Recompile the print version in order to recreate the configuration files for **lwarpmk**:
   Enter ⇒ `lwarpmk print`
3. Finally, recompile the HTML version with the new settings:
   Enter ⇒ `lwarpmk html`

Options for the lwarp package:
Use the following as options for `\usepackage[<options>]{lwarp}`:

Opt `HomeHTMLFilename`
Default: `\BaseJobname`

`HomeHTMLFilename`: Filename of the homepage, without the "`.html`" suffix. Defaults to the `\BaseJobname`. A common setting is:

```
HomeHTMLFilename=index
```

causing the homepage to be the file `index.html`. Underscores are allowed in `HomeHTMLFilename` and `HTMLFilename` options, but may need to be escaped elsewhere, such as when appearing in a list:

filename underscores

```
\item [\href{file\_name.pdf}{text}] \
```

See section 8.4.1 for examples of naming and numbering HTML files.

Opt `HTMLFilename`
Default: `<empty>`

`HTMLFilename`: A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty. See section 8.4.1 for examples of naming and numbering HTML files.

Opt `latexmk`
Default: `false`

`latexmk`: Controls whether **lwarp** uses **latexmk** to compile the document. This setting is written to **lwarpmk**'s configuration files.

Opt `mathsvg`
Default: `true`

`mathsvg`: Selects SVG display for math output. (The default.)

Opt `mathjax`
Default: `false`

`mathjax`: Selects MathJax for math output.

Opt `makeindex`
Default: `makeindex`

`makeindex`: Selects **makeindex** for index generation by **lwarpmk**.

Opt `xindy`
Default: `makeindex`

`xindy`: Selects **xindy** for index generation by **lwarpmk**.

Table 5: HTML settings

Option	Description
SideTOCDepth	Sectioning depth of the sideTOC.
FileDepth	Sectioning depth of the file splits.
CombineHigherDepths	Combine higher section levels.
FileSectionNames	Use section names for file names, else use numbers.
FootnoteDepth	Sectioning depth of footnotes.
\abstractname	The name of the abstract.
\mathimagename	The SVG math image lateximage alt tag.
\packagediagramname	The suffix for a package's lateximage alt tags.
\CSSFilename	The CSS for the following files.
\HTMLLanguage	The html lang tag.
\HTMLTitle	The HTML title meta tag, overriding \title.
\HTMLAuthor	The HTML author meta tag, overriding \author.
\HTMLDescription	The HTML description meta tag.
\HTMLFirstPageTop	Heading for the home page.
\HTMLPageTop	Heading for the other pages.
\HTMLPageBottom	Footing for all pages.

Placed in the preamble before `\begin{document}`:

Ctr `tocdepth`

`tocdepth`: Sectioning depth of the table of contents. See section 15 for a list of LaTeX stack depths.

Ctr `SideTOCDepth`
Default: `1`

`SideTOCDepth`: Sectioning depth of the sideTOC. Defaults to 1, causing the sideTOC to show sections but not subsections.

sideTOC

Each subpage of the website has its own small table of contents on the side (the "sideTOC"). Its depth is set by `SideTOCDepth`. This sideTOC is only shown if the web page is wide enough. When using a narrow web browser window, "responsive web design" is used to show the sideTOC at the top of the page and a link back to "Home" at the bottom.

It is recommended to set:

```
SideTOCDepth = FileDepth
```

or

```
SideTOCDepth = FileDepth+1
```

⚠ inaccessible pages

If `SideTOCDepth < FileDepth`, web pages will be inaccessible via the sideTOC.

Ctr `FileDepth`
Default: `-5`

`FileDepth`: Sectioning depth of file splits. Defaults to -5, causing the entire HTML website to be one single file.

- To place the entire file into one HTML page, use:
  `\setcounter{FileDepth}{-5}`
- To split the HTML file at `\section` depth, use:
  `\setcounter{FileDepth}{1}`
- ⚠ To ensure that the HTML pages/files are accessible:
  Place a `\tableofcontents` somewhere before the first section break (therefore in the "home page"), and set
  `tocdepth >= FileDepth`

Bool `CombineHigherDepths`
Default: `true`

`CombineHigherDepths`: Combine a higher section with its first lower subsections, down to the `FileDepth`. Defaults to true. Set to false to simulate the concept of a chapter opening on its own page, for example.

The file splits are controlled by the counter `FileDepth` and the boolean `CombineHigherDepths`. Setting `FileDepth` to 0 splits the file at chapters, 1 at sections, etc. `CombineHigherDepths` controls whether to combine pages at levels higher than the chosen `FileDepth`, such as in this tutorial where the page which opens the chapter also contains the first section.

⚠ Inaccesible pages!

Be careful to set `tocdepth` and `SideTOCDepth` to allow access to each page of the website. Set `tocdepth` and `SideTOCDepth` to be greater than or equal to `FileDepth`.

⚠ Lost in an old page!

When making changes to the file structure, it is possible to end up with the web browser pointing to an old file which is no longer in use. When this occurs, changes to the web site will not appear in the browser, even if reloading the page, because that page is no longer in use. It is best to return to the home page, clean the files (`lwarpmk cleanall`), change `FileDepth` and/or `CombineHigherDepths`, then finally recompile and renavigate to the desired page using the new file structure.

Bool `FileSectionNames`
Default: `true`

`FileSectionNames`: If true, web page filenames are derived from a sanitized version of the section names. If false, web pages are numbered. Either way, the `HTMLFilename` option is used as a prefix. See section 8.4.1 for examples of naming and numbering HTML files. The user must ensure that filenames are unique after begin sanitized. For example, math in the section name is removed before creating the filename, so the rest of the filename must be sufficiently unique to avoid name collisions.

⚠ Unique filename!

Ctr `FootnoteDepth`
Default: `3`

`FootnoteDepth`: Determines where to place pending footnotes. `3` places footnotes before each break down to the `\subsubsection` level. `1` places footnotes before each `\section` break. Any pending footnotes are also placed at the bottom of each page before each file break.

`\abstractname`
Default: `Abstract`

`\abstractname`: The name of the abstract. This may also be over-written by the **babel** package. Defaults to "Abstract".

Placed before `\begin{document}`, or before any sectioning command which causes a file break:

`\CSSFilename`
Default: `lwarp.css`

`\CSSFilename`: {⟨*filename.css*⟩} Sets the CSS file to use for the following files. May be changed before each each sectioning command which would cause a file split.

The CSS styles of the web pages are set by the `\CSSFilename` command. If `\CSSFilename` is not used, a default plain style is used to mimic printed LaTeX output. `lwarp_sagebrush.css` is a semi-fancy colored style as shown in this tutorial. Change it to `lwarp_formal.css` for a more formal look, or comment out the `\CSSFilename` command to see the default. `\CSSFilename` may be used before each file break to set the CSS for individual pagess of the website.

`\HTMLLanguage`
Default: `en-US`

`\HTMLLanguage`: {⟨*langauge*⟩} The HTML file's `html lang` meta tag. Defaults to `en-US`.

`\HTMLTitle`
Default: `\thetitle`

`\HTMLTitle`: {⟨*title*⟩} Overrides `\title` for the HTML header's meta title. Defaults to `\thetitle`, which is set by `\title`, or empty otherwise. Unlike the author, `\thetitle` is set by `\title` even if not using the **titling** package.

`\HTMLAuthor`
Default: `\theauthor`

`\HTMLAuthor`: {⟨*author*⟩} The HTML header's meta author. Defaults to

\theauthor, which is set by \author if using the **titling** package, but is empty otherwise. There are several ways to represent the author and affiliations, especially if using the **authblk** package, most of which do not result in a sensible \theauthor, so \HTMLAuthor is useful to create a list of authors without their affiliations.

\HTMLDescription
Default: <empty>

\HTMLDescription: {⟨*description*⟩} Sets the HTML description tag for the following files. May be changed before each each sectioning command which would cause a file split.

\HTMLFirstPageTop
Default: <empty>

\HTMLFirstPageTop: {⟨*contents*⟩} A user-definable custom action applied to the top of the home page. Useful for logos, etc. Defaults empty. Ignored in print output.

\HTMLPageTop
Default: <empty>

\HTMLPageTop: {⟨*contents*⟩} A user-definable custom action applied to the top of pages other than the home page. Useful for logos, etc. Defaults empty. \LinkHome may be used to place a link back to the homepage. Ignored in print output.

\HTMLPageBottom
Default: <empty>

\HTMLPageBottom: {⟨*contents*⟩} A user-definable custom action applied to the bottom of each web page. Useful for authors, copyright notices, contact information, etc. Defaults empty. \LinkHome may be used to place a link back to the homepage. Ignored in print output.

Placed in the home page before the first sectioning command which causes a file break:

\tableofcontents
⚠ TOC on the homepage!

\tableofcontents: Used to place a table of contents on the home page. This command must be used before the first file split, so that a way is available to navigate to other files from the homepage.

Links to each chapter/section are provided, as selected by tocdepth.

Placed in the document wherever necessary:

\mathimagename
Default: math image

\mathimagename: When creating an SVG math image, its alt tag may be set to the math expression, which may be hashed for image reuse. In the case of \ensuremath or after \StartDynamicMath, where the contents require a unique image for each instance of the same expression, the alt tag is set to \mathimagename, and the image is not reused.

This expression is visible in the browser if images are not loaded, and appears when the text is copied and pasted. The default is “math image”, and it may be changed according to the document’s language. This may be set in the preamble, or changed as necessary inside the document, where it will affect the following SVG math images.

\packagediagramname
Default: diagram

\packagediagramname: For many packages, the output is placed inside a `lateximage` with an HTML `alt` tag set to the package name followed by `\packagediagramname`. For example:

```
(-xy- diagram)
```

This expression is visible in the browser if images are not loaded, and appears when the text is copied and pasted. The default is "diagram", and may it be changed according to the document's language. This may be set in the preamble, or changed as necessary inside the document, where it will affect the following `lateximage`s.

Env warpprint

warpprint: An environment which is only used while generating print output. Place inside anything which does not apply to HTML and which may cause problems with **lwarp**. If **lwarp** knows about and emulates or supports a package then its related macros, lengths, counters, etc. probably won't have to be placed inside a `warpprint` environment, but unknown packages may cause problems which may be isolated from **lwarp** using this environment.

⚠ *Do not place anything else on the same line as `\end{warpprint}`.*

Env warpHTML

warpHTML: An environment which is only used while generating HTML output. This is useful for website logos and other items which have no purpose in printed output.

⚠ *Do not place anything else on the same line as `\end{warpHTML}`.*

\warpprintonly

\warpprintonly: {⟨*contents*⟩} A macro version of the `warpprint` environment.

\warpHTMLonly

\warpHTMLonly: {⟨*contents*⟩} A macro version of the `warpHTML` environment.

8.4.1 Example HTML file naming

Examples of ways to name or number HTML files:

Numbered HTML nodes:

Example: Homepage `index.html`, and `node-1`, `node-2`. [13]

```
\usepackage[
    HomeHTMLFilename=index,
    HTMLFilename={node-}
```

[13] See `\SetHTMLFileNumber` to number in groups by chapter, for example.

```
]{lwarp}
\boolfalse{FileSectionNames}
```

Named HTML sections, no prefix:

Example: `index.html`, and `About.html`, `Products.html`

```
\usepackage[
    HomeHTMLFilename=index,
    HTMLFilename={}
]{lwarp}
\booltrue{FileSectionNames}
```

Named HTML sections, with prefix:

Example: Homepage `mywebsite.html`, and additional pages such as `mywebsite-About.html`, `mywebsite-Products`, etc.

```
\usepackage[
    HomeHTMLFilename=mywebsite,
    HTMLFilename={mywebsite-}
]{lwarp}
\booltrue{FileSectionNames}
```

8.5 Customizing the CSS

`\CSSFilename`
Default: `lwarp.css`

`\CSSFilename` may be used to choose which `.css` file is used to display each page of the web site. Use `\CSSFilename` before `\begin{document}` to assign the style of the home page. If different parts of the website should have different styles, call `\CSSFilename` again before each section heading which creates a new file. This may be changed numerous times throughout the file, resulting in different HTML pages having different CSS files assigned:

```
...
\CSSFilename{myCSS.css}
\chapter{Another Chapter}
...
```

The styles provided by **lwarp** include:

`lwarp.css`: A default style if `\CSSFilename` is not used. This style is comparable to a plain LaTeX document. To set this style, you may use `\CSSFilename{lwarp.css}`, or no `\CSSFilename` call at all.

`lwarp_formal.css`: A formal style with a serif fonts and a traditional look.

`lwarp_sagebrush.css`: A style with muted colors, gradient backgrounds, additional borders, and rounded corners.

To see each style in use, change the `\CSSFilename` entry in the tutorial, `lwarpmk html` again, and then reload the tutorial webpage.

Custom css

A customized style may also be created. For each new project a file called `sample_project.css` is generated. This may be renamed to `<project>.css` then used by assigning `\CSSFilename{<project>.css}`.

⚠ Rename it!

Note that `sample_project.css` is overwritten whenever **lwarp** is loaded in print mode. It is therefore important to rename the file to something like `<project>.css` before using it, so that your own changes are not overwritten.

`<project>.css` has an entry which loads `lwarp.css`, and this entry may be changed to load `lwarp_formal.css` or `lwarp_sagebrush.css` if desired. Additional changes to the css may be made by making entries later in the `<project>.css` file.

File `lwarp.css`
File `project.css`
File `sample_project.css`

It is best to make a local project-specific css file such as `project.css`, containing only things which are different from `lwarp.css`. The file `project.css` should refer to `lwarp.css` as follows:

```
/* ( --- Start of project.css --- ) */
/* ( --- A sample project-specific CSS file for lwarp --- ) */

/* Uncomment one of the following: */
@import url("lwarp.css") ;
/* @import url("lwarp_formal.css") ; */
/* @import url("lwarp_sagebrush.css") ; */

/* Project-specific CSS setting follow here. */
/* . . . */

/* ( --- End of project.css --- ) */
```

Finally use `\CSSFilename{<project>.css}` in the document to activate the custom css.

8.6 Selecting the operating system

Prog `Unix`
Prog `Mac OS`
Prog `Linux`

lwarp tries to detect which operating system is being used. Unix / Mac OS / Linux

is the default (collectively referred to as "UNIX" in the configuration files), and MS-WINDOWS is supported as well.

Prog MS-Windows If MS-WINDOWS is not correctly detected, use the **lwarp** option `OSWindows`.

Prog Windows
Opt OSWindows When detected or specified, the operating-system path separator used by **lwarp** is modified, and the boolean `usingOSWindows` is set `true`. This boolean may be tested by the user for later use.

8.7 Selecting actions for print or HTML output

The following environments and macros are used to select actions which only apply to either traditional LaTeX print-formatted PDF generation, or to HTML generation.

For most of built-in LaTeX and many additional packages there is user-level source code support or emulation, so no special handling will be required. For those cases which **lwarp** does not handle by itself, the following environments and macros may be used to isolate sections of code for print-only or HTML-only.

These environments are also useful for creating a special version of the titlepage for print and another for HTML.

Env warpHTML Anything which is to be done only for HTML5 output is surrounded by a `warpHTML` environment:

```
\begin{warpHTML}
 ... something to be done only during HTML generation
\end{warpHTML}
```

⚠ `\end{warpHTML}` Do *not* place anything else on the same line as `\end{warpHTML}`. The exact phrase is used to mark the end of the environment.

Env warpprint Anything which is to be done only for print output is surrounded by a `warpprint` environment:

```
\begin{warpprint}
 ... something to be done only during traditional PDF generation
\end{warpprint}
```

As above, do not place anything else on the line with `\end{warpprint}`.

Env warpall Anything which is to be done for any output may be surrounded by a `warpall`

environment. Doing so is optional.

```
\begin{warpall}
 ... something to be done during print PDF or HTML output
\end{warpall}
```

As above, do not place anything else on the line with `\end{warpall}`.

Macros are also provided for print-only or HTML-only code:

`\warpprintonly` {⟨*actions*⟩}

Performs the given actions only when print output is being generated.

`\warpHTMLonly` {⟨*actions*⟩}

Performs the given actions only when HTML output is being generated.

8.8 Commands to be placed into the `warpprint` environment

Certain print-related commands should always be placed inside a `warpprint` environment, or may need other special handling. These are unrelated to HTML output, but are hard to isolate automatically. For example:

- Paragraph formatting: `\parindent` `\parskip`
- Manual page positions such as the **textpos** package, which is emulated but only in a limited way.

Some packages require additional setup commands. Where these packages are emulated for HTML, setup commands may work for the emulated HTML output as well as for print output. See the details for each package in this document for more information.

Also see section 13: Troubleshooting.

8.9 Title page

In the preamble, place an additional block of code to set the following:

```
\title{Document Title} % One line only
\author{Author One\affiliation{Affiliation One} \and
```

```
    Author Two\affiliation{Affiliation Two} }
\date{Optional date}
```

The title is used in the meta tags in the HTML files, unless overridden by `\HTMLTitle`, and the rest are used in `\maketitle`. To use a `\subtitle` or `\published` field, see section 62.8.

`\maketitle` Use `\maketitle` just after the `\begin{document}`, as this will establish the title of the homepage. Optionally, use a `titlepage` environment instead.

Env `titlepage` The `titlepage` environment may be used to hold a custom title page. The titlepage will be set in a `<div>` class `titlepage`, and `\printtitle`, etc. may be used inside this environment.

Env `titlingpage` Another form of custom title page, where `\maketitle` is allowed, and additional information may be included as well.

`\title` {⟨*title*⟩}

⚠ Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use a `\subtitle` command instead (section 62.8). The title will appear in the document `\maketitle` as a heading `<h1>`. The HTML meta `title` tag will also have this title, unless `\HTMLTitle` is used to set the meta title to something else instead.

`\author` {⟨*author*⟩}

⚠ In `\author`, use `\protect` before formatting commands such as `\textsc`. In HTML, the author will appear in a `<div>` of class `author` in the `\maketitle`. If the **titling** package is used, the author will also appear in a HTML meta tag, but `\HTMLAuthor` may be necessary to create a plain list of names if `\author` had affiliations added. `\affiliation` is a new addition to **lwarp**.

`\date` {⟨*date*⟩}

`\date` works as expected. In HTML, this will appear in a `<div>` class `titledate`.

`\thanks` {⟨*text*⟩}

`\thanks` are allowed in the titlepage fields, and will be rendered as HTML notes at the bottom of the title page.

8.10 HTML page meta descriptions

`\HTMLDescription` {⟨*A description of the web page.*⟩} The default is no description.

limitations

Each page of HTML output should have its own HTML meta description, which usually shows up in web search results, is limited to around 150 characters in length, and should not include the ASCII double quote character (").

placement

Use `\HTMLDescription` just before `\begin{document}` to set the description of the home page, and also just before each sectioning command such as `\chapter` or `\section` where a new file will be generated, depending on `FileDepth`. For example, if `FileDepth` is 1, use `\HTMLDescription` just before each `\section` command, and that description will be placed inside the HTML page for that `\section`. The same descrition will be used for all following HTML files as well, until reset by a new `\HTMLDescription`. It is best to use a unique description for each HTML file.

disabling

To disable the generation of HTML description meta tags, use:

```
\HTMLDescription{}
```

8.11 HTML page meta title

`\HTMLTitle` `{`⟨*title*⟩`}` Sets the contents of the web page `<meta name="title">` element. Defaults to `\HTMLtitle{\thetitle}`. May be set empty to cancel the meta title tag.

8.12 HTML page meta author

`\HTMLAuthor` `{`⟨*author*⟩`}` Sets the contents of the web page `<meta name="author">` element. Defaults to `\HTMLAuthor{\theauthor}`. May be set empty to cancel the meta author tag.

`\author` may be used to create a list of authors and their affiliations, in several formats if using **authblk**, and these may not successfully parse properly into a sensible list for `\theauthor`. `\HTMLAuthor` may be used to set the meta tag to a simple list of names.

9 Special cases and limitations

Some commonly-used LaTeX expressions should be modified as follows to allow for a smooth conversion to both HTML and print-formatted outputs.

Need help?

The index to this document contains several hundred custom entries. Also included are automated entries for each package, macro, environment, counter, boolean, and other objects; individually and also sorted by category. A Troubleshooting section is also available.

9.1 Things to avoid

In the document, avoid the following:

`page` counter: Do not adjust the `page` counter. If doing so is required for the print version, place the adjustment inside a `warpprint` environment.

Custom math environment macros: Do not use expressions such as `\beq` as a replacement for `\begin{equation}`.

Custom macros in section, figure, table names: Custom macros which appear in sectioning commands or float captions then appear in the `.toc`, `.lof`, and `.lot` lists, and should be made robust using `\newrobustcmd` or `\robustify` from **etoolbox**, **xparse**, etc.

When setting `FileSectionNames` to `true` to name the HTML files from the section names, the file names are created from sanitized versions of the chapter or section names, but the section names must be plain text or something which expands into plain text. Robust macros will not work at the sectioning level which is used for file names, but a robust macro or other complicated name may be used for the manditory argument of `\chapter`, `\section`, etc., if a plain-text version is also included in the optional argument:

```
\chapter[Plain Name]{\ARobustMacro{Fancy Name}}
```

9.2 Formatting

9.2.1 Text formatting

⚠ `\bfseries`, etc. `\textbf`, etc. are supported, but `\bfseries`, etc. work only in some situations.

⚠ HTML special chars `&`, `<`, and `>` have special meanings in HTML. If `\&`, `\textless`, and `\textgreater` are used, the proper result should occur in HTML, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

9.2.2 Horizontal space

`\hspace` `\hspace` is converted to an inline HTML span of the given width, except that `0` width is ignored, a width of `.16667em` is converted to an HTML thin breakable space (`U+2009`), and a `\fill` is converted to a `\qquad`.

`\,` `~` and `\,` are converted to HTML entities.

\kern
\hskip

\kern and \hskip are entered into the HTML PDF output as-is, then interpreted by **pdftotext**, and thus usually appear as a single space.

9.2.3 Text alignment

Use the environments `center`, `flushright`, `flushleft` instead of the macros `\centering`, `\raggedright`, `\raggedleft`.

figure & table alignment

`\centering`, etc. are honored in a `figure` or `table` if they are the first command inside the float:

```
\begin{table*}
\centering
\caption{A Table}
...
```

9.2.4 Accents

Native LaTeX accents such as `\"` will work, but many more kinds of accents are available when using Unicode-aware XeLaTeX and LuaLaTeX.

9.2.5 textcomp package

Pkg textcomp

Some **textcomp** symbols do not have Unicode equivalents, and thus are not supported.

⚠ missing symbols

Many **textcomp** symbols are not supported by many fonts. Try using more complete fonts in the CSS, but expect to see gaps in coverage.

9.2.6 Superscripts and other non-math uses of math mode

Use `\textsuperscript{x}` instead of `$^{x}$`

9.2.7 Empty `\item` followed by a new line of text or a nested list:

lists

Use a trailing backslash: `\item[label] \`

9.2.8 Filenames and URLs in lists or footnotes

filename underscore

Escape underscores in the filenames:

```
\item[\href{file\_name.pdf}{text}]
```

9.2.9 relsize package

Pkg relsize

For HTML only the inline macros are supported: `\textlarger`, `\textsmaller`, and `\textscale`. Each becomes an inline span of a modified `font-size`.

`\relsize`, `\larger`, `\smaller`, and `\relscale` are ignored.

While creating SVG math for HTML, the original definitions are temporarilty restored, and so should work as expected.

⚠ not small

The HTML browser's setting for minumum font size may limit how small the output will be displayed.

9.3 Boxes and minipages

9.3.1 Marginpars

`\marginpar` `[`⟨*left*⟩`] {`⟨*right*⟩`}`

`\marginpar` may contains paragraphs, but in order to remain inline with the surrounding text **lwarp** nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `<br />` tags.

`\marginparBlock` `[`⟨*left*⟩`] {`⟨*right*⟩`}`

To include block-related macros, use `\marginparBlock`, which takes the same arguments but creates a `<div>` instead of a `<span>`. A line break will occur in the text where the `\marginBlock` occurs.

9.3.2 Save Boxes

LaTeX boxes are placed inline and do not allow line breaks, so boxes with long contents may overflow the line during HTML conversion. This is mostly a problem when the boxes contain objects which themselves hold large HTML tags, such as rotation commands with long contents. When this object overflows the line, some HTML code will be lost and the page will be corrupted.

9.3.3 Minipages

⚠ inline

A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made `inline-block`.

placement

Minipages and parboxes will be placed side-by-side in HTML unless you place a `\newline` between them.

side-by-side

Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML. Paragraph tags are surpressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

in a span

There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an "inline" format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.

size

When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.

no-width minipages

A minipage of width exactly `\linewidth` is automatically given no HTML width.

full-width minipages

A new macro `\minipagefullwidth` requests that the next minipage be generated without an HTML `width` attribute, allowing it to be the full width of the display rather than the fixed width given.

⚠ text alignment

Nested minipages adopt their parent's text alignment in HTML, whereas in regular LaTeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

9.3.4 Side-by-side minipages

Place side-by-side minipages inside a `center` environment, with horizontal space between them, such as `\quad`, `\qquad`, `\hspace`, or `\hfill`. The result is similar in print and HTML. Do not use space commands at the start or end of the line.

9.3.5 Framed minipages and other environments

`\fbox` can only be used around inline `<span>` items during HTML output, but HTML cannot place a block element such as a `<div>` for a minipage or a list inside of a `<span>`. Several options are provided for framing an object, depending on which kind of object and which packages are loaded:

\fbox
\fboxBlock
Env fminipage

For a framed object, options include:

To remove the frame in HTML output: Place the \fbox command and its closing brace inside warpprint environments. This will nullify the frame for HTML output.

For inline text:

To frame the contents inline with some formatting losses in HTML: This is the default action of \fbox when enclosing a minipage. During HTML output, \fbox nullifies the HTML tags for minipage, \parbox, and lists. The contents are included as inline text inside the \fbox's of class framebox. For lists, line breaks are converted to HTML breaks. The result is a plain-text inline version of the contents, framed inline with the surrounding text, but lacking any extra HTML markup.

For inline minipage and lists:

To frame the contents on their own line with improved formatting in HTML: A new command \fboxBlock is included, intended to be a direct replacement for \fbox for cases where the \fbox surrounds a minipage, table, or list. For print output, this behaves as \fbox. For HTML output, the contents are placed inside an HTML with the class framed, resulting in the contents being placed on their own line with a frame surrounding them. The contents preserve their HTML formatting, so lists and minipages look nicer, and valid HTML is created for a tabular. While an \fbox containing a tabular is valid LaTeX code, the result in HTML is problematic since a table is a not a , so use \fboxBlock around a tabular, or else place the tabular inside a minipage, or use fminipage, described next. Also see below regarding the "Misplaced alignment tab character &" error.

For display tabular, minipages, and lists:

To create a framed minipage in both print and HTML: A new environment fminipage is included. For print output, this is identical to minipage, except that it is also framed. For HTML output, this forms a of class framed, the contents preserve their HTML formatting, and valid HTML is created for a tabular. Also see below regarding the "Misplaced alignment tab character &" error.

colored boxes and frames:

To create colored frames and boxes: See section 409 for **xcolor**'s \colorbox and \fcolorbox, and **lwarp**'s additional \colorboxBlock and \fcolorboxBlock.

⚠ Misplaced alignment tab character &

To frame tables or verbatim environments: Place the contents inside a fminipage, or perhaps a \fboxBlock for a tabular. Also, if using \fboxblock with tabular, you will have to use \StartDefiningTabulars before the start of the macro which uses \fboxBlock and the tabular, and \StopDefiningTabulars afterwards. Also see the **lwarp** documentation for the **fancybox** package.

To frame equations: See section 189 for the **fancybox** package.

For fancy framed minipages: See packages **boxedminipage**, **shadow**, **fancybox**, **framed**, **mdframed**.

Custom environments: Use a custom environment to create a sidebar, containing a `BlockClass` environment with custom css formatting, and `\warpprintonly{\hrule}` command:

```
\begin{BlockClass}{frameminipage}% ignored in print output
  % use CSS to format div class ``framedminipage''
\warpprintonly{\hrule} % only appears in print output
Contents
\warpprintonly{\hrule} % only appears in print output
\end{BlockClass}
```

9.3.6 fancybox package

Pkg `fancybox`

framed equation example

fancybox's documentation has an example `FramedEqn` environment which combines math, `\Sbox`, a minipage, and an `\fbox`. This combination requires that the entire environment be enclosed inside a `lateximage`, which is done by adding `\lateximage` at the very start of `FramedEqn`'s beginning code, and `\endlateximage` at the very end of the ending code. Unfortunately, the HTML `alt` attribute is not used here.

```
\newenvironmentFramedEqn
{
\lateximage% NEW
\setlength{\fboxsep}{15pt}
...}{...
\[\fbox{\TheSbox}\]
\endlateximage% NEW
}
```

framing alternatives

`\fbox` works with **fancybox**. Also see **lwarp**'s `\fboxBlock` macro and `fminipage` environment for alternatives to `\fbox` for framing environments.

framed table example

The **fancybox** documentation's example framed table using an `\fbox` containing a `tabular` does not work with **lwarp**, but the `FramedTable` environment does work if `\fbox` is replaced by `\fboxBlock`. This method loses HTML formatting. A better method is to enclose the table's contents inside a `fminipage` environment. The caption may be placed either inside or outside the `fminipage`:

```
\begin{table}
\begin{fminipage}{\linewidth}
\begin{tabular}{lr}
...
\end{tabular}
\end{fminipage}
\end{table}
```

framed verbatim

lwarp does not support the `verbatim` environment inside a span, box, or **fancybox**'s `\Sbox`, but a `verbatim` may be placed inside a `fminipage`. The **fancybox** documentation's example `FramedVerb` may be defined as:

```
\newenvironment{FramedVerb}[1] % width
{
\VerbatimEnvironment
\fminipage{#1}
\beginVerbatim
}{
\endVerbatim
\endfminipage
}
```

framed `\VerbBox`

fancybox's `\VerbBox` may be used inside `\fbox`.

indented alignment

`LVerbatim`, `\LVerbatimInput`, and `\LUseVerbatim` indent with horizontal space which may not line up exactly with what **pdftotext** detects. Some lines may be off slightly in their left edge.

9.3.7 mdframed package

Pkg `mdframed` support

Most basic functionality is supported, including frame background colors and single-border colors and thickness, title and subtitle background colors and borders and thickness, border radius, and shadow. CSS classes are created for **mdframed** environments and frame titles.

⚠ loading

When used, **lwarp** loads **mdframed** in HTML with `framemethod=none`.

font

For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the **mdframed** source). Since **lwarp** does not support `\bfseries` and friends, only one font selection may be made at a time.

theoremtitlefont

`theoremtitlefont` is not supported, since the following text is not in braces in the **mdframed** source.

footnotes

Footnotes are currently placed at the bottom of the HTML page.

ignored options

`userdefinedwidth` and `align` are currently ignored.

9.4 Cross-references

labels
⚠ underscores

Labels with special characters may be a problem. It is best to stick with alphanumeric, hyphen, underscore, and perhaps the colon (if not French).

`\nameref`
⚠ empty link

`\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.

9.4.1 Page references

⚠ LaTeX page numbers

The printed page does not translate to the HTML page, so `\pageref` references are converted to parentheses containing `\pagerefPageFor`, which defaults to "see ", followed by a hyperlink to the appropriate object.

Ex:

```
\ref{sec:name} on page \pageref{sec:name}
```

in HTML becomes:

"Sec. 1.23 on page (**see** sec. 1.23)".

`\pagerefPageFor` may be redefined to "page for ", empty, etc. See page 461.

9.4.2 cleveref and varioref packages

Pkg `cleveref`
Pkg `varioref`
⚠ **cleveref** page numbers

cleveref and **varioref** are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to "for".

Ex:

```
\cpageref{tab:first,tab:second}
```

in HTML becomes:

"pages **for** table 4.1 and **for** table 4.2"

See `\cpagerefFor` at page 539 to redefine the message which is printed for page number references.

9.4.3 Hyperlinks, hyperref, and url

Pkg `hyperref`
Pkg `url`

lwarp emulates **hyperref**, including the creation of active hyperlinks, but does not

require that **hyperref** be loaded by the document.

⚠ % Do not place a comment with a % character between arguments for `\hyperref`, etc., as it is neutralized for inclusion in HTML URLs.

lwarp can also load **url**, but **url** should not be used at the same time as **hyperref**, since they both define the `\url` command. **lwarp** does not (yet) attempt to convert **url** links into hyperlinks during HTML output, nor does **url** create hyperlinks during print output.

⚠ backref When generating HTML, **lwarp**'s emulation of **hyperref** does not automatically load **backref**, so **backref** must be loaded explicitly.

9.4.4 Footnotes and page notes

lwarp uses native LaTeX footnote code, although with its own `\box` to avoid the LaTeX output routine. The usual functions mostly work as-is.

The **footmisc** `stable` option is emulated by **lwarp**.

⚠ sectioning commands When using footnotes in sectioning commands, to generate consistent results between print and HTML, use the **footmisc** package with the `stable` option, provide a short TOC entry, and `\protect` the `\footnote`:

```
\usepackage[stable]{footmisc}
...
\subsection[Subsection Name]
    {Subsection Name\protect\footnote{A footnote.}}
```

If using **memoir** class, with which **lwarp** preloads **footmisc**, the `stable` option must be declared before **lwarp** is loaded:

```
\PassOptionsToPackage{stable}{footmisc}
\usepackage{lwarp}
...
```

Do not use a starred sectioning command. As an alternative, it may be possible to adjust `\secnumdepth` instead.

⚠ `\VerbatimFootnotes` ⚠ sectioning or displaymath If using **fancybox** or **fancyvrb** with `\VerbatimFootnotes`, and using footnotes in a sectioning command or display math, use `\footnotemark` and `\footnotetext`:

```
\subsection[Subsection Name]
    {Subsection Name\protect\footnotemark}
\footnotetext{A footnote with \verb+verbtim+.}
```

and likewise for equations or display math.

At present there is a bug such that paragraph closing tags are not present in footnotes when \VerbatimFootnotes are selected. The browser usually compensates.

⚠ **pfnote** numbers

While emulating **pfnote**, **lwarp** is not able to reset HTML footnote numbers per page number to match the printed version, as HTML has no concept of page numbers. **lwarp** therefore uses continuous footnote numbering even for **pfnote**.

9.5 Front and back matter

9.5.1 Custom classes with multiple authors and affiliations

Some classes allow multiple authors and affiliations. Often it is possible to emulate these using a standard class along with **authblk**:

```
%\documentclass{customclass} % for print document
\documentclass{article} % for HTML document

\usepackage{lwarp}
\begin{warpHTML}
\usepackage{authblk}
\let\affiliation\affil % maybe required
\end{warpHTML}
```

9.5.2 Starred chapters and sections

The following describes \ForceHTMLPage and \ForceHTMLTOC, which may be used for **endnotes**, **glossaries**, **tocbibind**, and the index. See the following sections where applicable. Continue here if interested in the reason for adding these commands to **lwarp**.

Some packages use \chapter* or \section* to introduce reference material such as notes or lists, often to be placed in the back matter of a book. These starred sections are placed inline instead of on their own HTML pages, and they are not given TOC entries.

lwarp provides a method to cause a starred section to be on its own HTML page, subject to FileDepth, and also a method to cause the starred section to have its own TOC entry during HTML output.

\ForceHTMLPage

To place a starred section on its own HTML page, use \ForceHTMLPage just before the \chapter* or \section*. **lwarp** will create a new page for the starred sectional unit.

A starred sectional unit does not have a TOC entry unless one is placed manually. The typical method using \phantomsection and \addcontentsline works for inline text but fails when the new starred section is given its own webpage after the TOC entry is created, or when creating an EPUB where the TOC entry will point to the page before the starred section. If the starred section has its own HTML page but no correct TOC entry pointing to that page, the page will be inaccessible unless some other link is created.

⚠ inaccessible HTML page

\ForceHTMLTOC

To automatically force the HTML version of the document to have a TOC entry for a starred section, use \ForceHTMLTOC just before the \chapter* or \section*, and place \phantomsection and \addcontentsline inside a warpprint environment.

For print output, \ForceHTMLTOC and \ForceHTMLPage have no effect.

9.5.3 abstract package

Pkg abstract

⚠ missing TOC

If using the number option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a file split, which would put a table of contents out of the home page if it is after the abstract.

9.5.4 titling and authblk

Pkg titling

Pkg authblk

package support

⚠ load order

\published and \subtitle

lwarp supports the native LaTeX titling commands, and also supports the packages **authblk** and **titling**. If both are used, **authblk** should be loaded before **titling**.

If using the **titling** package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 62.8.

9.5.5 tocloft package

Opt tocloft titles

Pkg tocloft

Pkg tocloft

⚠ **tocloft** & other packages

If using **tocloft** with **tocbibind**, **anonchap**, **fncychap**, or other packages which change chapter title formatting, load **tocloft** with its titles option, which tells **tocloft** to use standard LaTeX commands to create the titles, allowing other packages to work with it.

9.5.6 appendix package

Pkg appendix

⚠ incorrect TOC link

During HTML conversion, the option toc without the option page results in a TOC link to whichever section was before the appendices environment. It is recommended to use both toc and also page at the same time.

9.5.7 pagenote package

Pkg pagenote

pagenote works as-is, but the page option is disabled.

9.5.8 endnotes package

Pkg endnotes

table of contents

To place the endnotes in the TOC, use:

```
\usepackage{endnotes}
\appto\enoteheading{\addcontentsline{toc}{section}{\notesname}}
\renewcommand*{\notesname}{Endnotes} % optional
```

HTML page

To additionally have the endnotes on their own HTML page, if `FileDepth` allows:

```
\ForceHTMLPage
\theendnotes
```

9.5.9 BibTeX

`\etalchar`

Displays a superscript "+" to indicate "and others".

⚠ Modify *.bib

When enough authors are cited for a source, BibTeX may use the `\etalchar` command to display a math superscript with a + character to indicate "and others". Without modification, this will result in an "Improper `\prevdepth`" error. At present, **lwarp** requires that `\etalchar` be replaced by a text superscript. To do so, add to the start of the `.bib` file the following:

```
@PREAMBLE{"\let\etalchar\relax \newcommand{\etalchar}[1]{\textsuperscript{#1}}"}
```

9.5.10 glossaries package

Pkg glossaries

processing glossaries

Opt GlossaryCmd

Default: `makeglossaries`

Opt `lwarpmk` printglossary

Opt `lwarpmk` htmlglossary

lwarpmk has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary`, which process the glossaries created by the **glossaries** package using that package's ***makeglossaries*** command.

The shell command to execute is set by the **lwarp** option `GlossaryCmd`, which defaults to ***makeglossaries***. The print or HTML glossary filename is appended to this command.

⚠ ***makeglossaries*** not found

In some situations it may be required to modify the default command, such as to add the ***perl*** command in front:

```
\usepackage[
   GlossaryCmd={perl makeglossaries},
] {lwarp}
```

xindy language

To set the language to use for processing glossaries with **xindy**:

```
\usepackage[
   GlossaryCmd={makeglossaries -L english},
] {lwarp}
```

Other options for ***makeglossaries*** may be set as well.

placement and TOC options

The glossaries may be placed in a numbered or unnumbered section, given a TOC entry, and placed inline or on their own HTML page:

Numbered section, on its own HTML page:

```
\usepackage[xindy,toc,numberedsection=nolabel]{glossaries}
...
\printglossaries
```

Unnumbered section, inline with the current HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\printglossaries
```

Unnumbered section, on its own HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\ForceHTMLPage
\printglossaries
```

⚠ glossary style

The default `style=item` option for **glossaries** conflicts with **lwarp**, so the style is forced to `index` instead.

⚠ number list

The page number list in the printed form would become `\namerefs` in HTML, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

print/HTML versions

The print and HTML versions of the glossary differ in their internal page numbers. Separate commands for generating print and HTML glossaries are used, even though the page number is currently ignored.

9.5.11 Indexing overview

There are many ways to process indexes for a LaTeX document, including native LaTeX capabilities, a number of packages and classes, the possible availability of shell escape

and **latexmk**, and the need to process print and HTML versions. **lwarp** attempts to provide easy recompilation of indexes along with the rest of the document, but the various indexing options must be set correctly. Numerous examples are given below. Some differ in minor details, so the important parts are highlighted in red, and options are in green.

Once set up properly, the entire document may be recompiled with ***lwarpmk print*** and ***lwarpmk html***. In some cases, it will also be necessary to compile the indexes with ***lwarpmk printindex*** and ***lwarpmk htmlindex***. A recompile may then be forced with ***lwarpmk print1*** and ***lwarpmk html1***.

manual processing

The user may continue to process indexes manually or by shell script without the use of **lwarpmk**, but adjustments will be required to process HTML indexes as well. In general, `*.idx` and `*.ind` files will be accompanied by `*_html.idx` and `*_html.ind` files.

custom index style

If using a custom indexing style file, see sections 9.5.17 and 9.5.18.

source code

See section 72 for **lwarp**'s core index and glossary code, section 245 for **index**, section 356 for **splitidx**, section 243 for **imakeidx**, section 382 for **tocbibind**, and section 422.17 for **memoir**'s indexing patches.

9.5.12 Indexing with basic LaTeX and makeidx

lwarpmk processing

The following allow the user to process indexes automatically, or using **lwarpmk**'s commands:

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

For a single index using makeindex:

```
\usepackage[makeindex,latexmk] {lwarp}
```

The usual `.idx` and `.ind` files will be used, along with the new `lwarp.ist` style file. When creating the HTML index, "`_html`" is automatically appended to each of the names.

lwarpmk will use **latexmk** if specified, in which case **latexmk** will create the index automatically. Otherwise, use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

For a single index using xindy:

```
\usepackage[
    xindy,
    xindyLanguage=english,                    <optional>
    xindyCodepage=utf8,                       <optional>
    latexmk                                   <optional>
]{lwarp}
```

The usual `.idx` and `.ind` files will be used, along with the new `lwarp.xdy` style file.

lwarpmk will use **latexmk** if specified, in which case **latexmk** will create the index automatically. Otherwise, use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

9.5.13 Indexing with index

lwarp is told how to use **makeindex** using the `PrintIndexCmd` and `HTMLIndexCmd` options. The file `lwarp.ist` is specified, which generates index letter heads for print output and also allows special HTML formatting for HTML output.

For multiple indexes using makeindex and index:

(Assuming that the second index has file extensions `.sist` and `.sind`)

```
\usepackage[
  makeindex, latexmk,
  PrintIndexCmd={
    makeindex -s lwarp.ist <projectname>.idx ;
    makeindex -s lwarp.ist
      -o <projectname>.sind <projectname>.sidx
  },
  HTMLIndexCmd={
    makeindex -s lwarp.ist <projectname>_html.idx ;
    makeindex -s lwarp.ist
      -o <projectname>_html.sind <projectname>_html.sidx
  }
]{lwarp}
\usepackage{index}
...
\makeindex
\newindex{secondname}{sidx}{sind}{Second Index}
```

⚠ WINDOWS *For WINDOWS, replace the two ";" characters with "&".*

When creating the HTML index, "_html" is automatically appended to the index filenames.

Use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

If the `latexmk` option is selected for **lwarp**, **latexmk** will compile the document but will *not* compile the indexes. ***lwarpmk printindex*** and ***lwarpmk htmlindex*** will still be required.

9.5.14 Indexing with splitidx

lwarp is told how to use **splitindex** using the `PrintIndexCmd` and `HTMLIndexCmd` options. The file `lwarp.ist` is specified, which generates index letter heads for print output and also allows special HTML formatting for HTML output.

If the `latexmk` option is selected for **lwarp**, **latexmk** will compile the document but will *not* compile the indexes. ***lwarpmk printindex*** and ***lwarpmk htmlindex*** will still be required.

⚠ `\thepage` When using `\AtWriteToIndex` or `\AtNextWriteToIndex`, the user must not refer to `\thepage` during HTML output, as the concept of a page number is meaningless. Instead, do

```
\addtocounter{LWR@autoindex}{1}
\LWR@new@label{LWRindex-\arabic{LWR@autoindex}}
```

where the `\index`-like action occurs, and then refer to `\arabic{LWR@autoindex}` instead of `\thepage` where the reference should occur.

See section 422.17 in the **lwarp-patch-memoir** package for the `\@@wrspindexhyp` macro as an example.

For multiple indexes using makeindex and splitidx:

```
\usepackage[
  makeindex, latexmk,
  PrintIndexCmd={
    splitindex <projectname> -- -s lwarp.ist
  },
  HTMLIndexCmd={
    splitindex <projectname>_html -- -s lwarp.ist
  }
]{lwarp}
\usepackage{splitidx}
...
\makeindex
\newindex[Second Index]{secondname}
```

When creating the HTML index, "_html" is automatically appended to each of the names.

Use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

For multiple indexes using xindy and splitidx:

```
\usepackage[
  xindy, latexmk,
  PrintIndexCmd={
    splitindex -m xindy <projectname> -- -M lwarp.xdy
      -L english -C utf8                          <optional>
  },
  HTMLIndexCmd={
    splitindex -m xindy <projectname>_html -- -M lwarp.xdy
      -L english -C utf8                          <optional>
  }
]{lwarp}
\usepackage{splitidx}
...
\makeindex
\newindex[Second Index]{secondname}
```

When creating the HTML index, "_html" is automatically appended to each of the names.

Use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

9.5.15 Indexing with imakeidx

Due to the number of methods which may be used to process multiple indexes, the options for style file and **xindy** language and codepage must be specified in one of several different ways. These are described in detail later in this section, but are summarized here.

If shell escape is used, **imakeidx** will automatically compile the indexes by itself. Options specifying a custom style file and **xindy** language and codepage must be specified for each `\makeindex` command using its `options=` option, which must include **lwarp**'s special `lwarp.ist` or `lwarp.xdy` file, or a file based on them. If using a custom indexing style file, see sections 9.5.17 and 9.5.18. The `splitindex` option is also available of shell escape is used, in which case the **splitidx** package and **splitindex** program will also be used.

If shell escape is not possible, **latexmk** may be used to automatically compile the indexes. The style, language, and codepage options are specified with **lwarp**'s `makeindexStyle`, `xindyStyle`, `xindyLanguage`, and `xindyCodepage` options. These are passed to **latexmk** by **lwarpmk**'s ***lwarpmk printindex*** and ***lwarpmk htmlindex*** commands.

Where shell escape and **latexmk** are not possible, **lwarpmk** may be used to manually compile the indexes. **lwarp**'s `PrintIndexCmd` and `HTMLIndexCmd` options are used.

For a single or multiple indexes using makeindex and imakeidx:

The index style `lwarp.ist` is automatically used for HTML output. This file turns on letter headings, so it may be desirable to specify it as an option, in which case it will also be used for print output, which will help match the print and HTML output.

```
\usepackage[makeindex,latexmk] {lwarp}
\usepackage[makeindex]{imakeidx}
...
\makeindex[options={-s lwarp.ist}]
\makeindex[name=secondname,options={-s lwarp.ist}]
```

imakeidx will automatically compile the indexes. Shell escape is not required while using **makeindex**. `latexmk` may be specified, and if so it will be used for ***lwarpmk print*** and ***lwarpmk html***, but **imakeidx** will actually create the indexes.

For a single or multiple indexes using makeindex and splitindex with imakeidx:

The index style `lwarp.ist` is automatically used for HTML output. This file turns on letter headings, so it may be desirable to specify it as an option, in which case it will also be used for print output, which will help match the print and HTML output.

```
\usepackage[makeindex,latexmk] {lwarp}
\usepackage[makeindex,splitindex]{imakeidx}
...
\makeindex[options={-s lwarp.ist}]
\makeindex[name=secondname,options={-s lwarp.ist}]
```

⚠ enable shell escape

Shell escape is required while using **splitindex**. For the first compile, use

```
Enter ⇒  pdflatex -shell-escape projectname.tex
Enter ⇒  pdflatex --enable-write18 projectname.tex    (MikTeX)
```

or similar with **xelatex** or **lualatex**. **lwarp** will remember that shell escape was used.

imakeidx will automatically execute **splitindex**, and will also use **makeindex** to compile the indexes.

`latexmk` may be specified, and if so it will be used for ***`lwarpmk print`*** and ***`lwarpmk html`***, but **imakeidx** will actually create the indexes.

For multiple indexes using xindy and imakeidx, using shell escape:

Options may be given to **imakeidx**'s `\makeindex` command. The style file `lwarp.xdy` is automatically used for HTML output, and is not necessary for print output since the output will be similar. If language or codepage must be set, they should be specified as options for `\makeindex`, since **imakeidx** will process the indexes.

```
\usepackage[xindy,latexmk] {lwarp}
\usepackage[xindy,splitindex]{imakeidx}
...
\makeindex[
  options={ -M lwarp.xdy -L english -c utf8 }
]
\makeindex[
  name=secondname,
  options={ -M lwarp.xdy -L english -c utf8 }
]
```

⚠ enable shell escape

For the first compile, use

```
Enter ⇒  pdflatex -shell-escape projectname.tex
Enter ⇒  pdflatex --enable-write18 projectname.tex    (MikTeX)
```

or similar with **xelatex** or **lualatex**. **lwarp** will remember that shell escape was used.

imakeidx will automatically execute **splitindex** if selected, and will also use **xindy** to compile the indexes.

If selected, **latexmk** will automatically recompile the entire document as necessary.

For indexes using xindy and imakeidx, without shell escape, but *with* latexmk:

lwarp's options are used, and are passed to **latexmk**.

```
\usepackage[
  xindy,
  xindyLanguage=english,                    <optional>
  xindyCodepage=utf8,                       <optional>
  latexmk,
]{lwarp}
\usepackage[xindy]{imakeidx}
...
\makeindex
\makeindex[name=secondname]
```

latexmk will create the indexes automatically when ***lwarpmk print*** and ***lwarpmk html*** are executed.

For indexes using xindy and imakeidx, without shell escape, and *without* latexmk:

lwarpmk must be told how to create the indexes:

```
\usepackage[
  xindy,
  PrintIndexCmd={
    xindy -M lwarp.xdy -L english -C utf8
      <projectname>.idx ;
    xindy -M lwarp.xdy -L english -C utf8
      secondname.idx
  },
  HTMLIndexCmd={
    xindy -M lwarp.xdy -L english -C utf8
      <projectname>_html.idx ;
    xindy -M lwarp.xdy -L english -C utf8
      secondname_html.idx
  }
]{lwarp}
\usepackage[xindy]{imakeidx}
...
\makeindex
\makeindex[name=secondname]
```

⚠ WINDOWS *For WINDOWS, replace the two ";" characters with "&".*

`<projectname>` is the `\jobname`: if compiling "`name.tex`", use the filenames `name.idx` and `name_html.idx`.

Use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

9.5.16 Indexes with memoir

For a single index with memoir and makeindex:

```
\documentclass{memoir}
\usepackage[makeindex,latexmk]{lwarp}
...
\makeindex
```

The usual `.idx` and `.ind` files will be used, along with the `lwarp.ist` style file.

lwarpmk will use **latexmk** if specified, in which case **latexmk** will create the index automatically. Otherwise, use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

For multiple indexes with memoir and makeindex, using latexmk:

lwarp's options are used, and are passed to **latexmk**.

```
\documentclass{memoir}
\usepackage[makeindex,latexmk]{lwarp}
...
\makeindex
\makeindex[secondname]
```

lwarpmk will use **latexmk** to create the indexes automatically when the user executes ***lwarpmk print*** and ***lwarpmk html***.

For multiple indexes with memoir and makeindex, *without* latexmk:

lwarpmk must be told how to create the indexes:

```
\documentclass{memoir}
\usepackage[
  makeindex,
  PrintIndexCmd={
    makeindex -s lwarp.ist <projectname>.idx ;
    makeindex -s lwarp.ist secondname.idx
  },
  HTMLIndexCmd={
    makeindex -s lwarp.ist <projectname>_html.idx ;
    makeindex -s lwarp.ist secondname_html.idx
  }
]{lwarp}
...
\makeindex
\makeindex[secondname]
```

⚠ WINDOWS *For WINDOWS, replace the two ";" characters with "&".*

`<projectname>` is the `\jobname`: if compiling "`name.tex`", use the filenames `name.idx` and `name_html.idx`.

Use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

For a single index with memoir and xindy:

```
\documentclass{memoir}
\usepackage[
    xindy,
    xindyLanguage=english,                    <optional>
    xindyCodepage=utf8,                       <optional>
    latexmk                                   <optional>
]{lwarp}
...
\xindyindex
\makeindex
```

The usual `.idx` and `.ind` files will be used, along with the `lwarp.xdy` style file.

lwarpmk will use **latexmk** if specified, in which case **latexmk** will create the index automatically. Otherwise, use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

For multiple indexes with memoir and xindy, using latexmk:

lwarp's options are used, and are passed to **latexmk**.

```
\documentclass{memoir}
\usepackage[
    xindy,
    xindyLanguage=english,                    <optional>
    xindyCodepage=utf8,                       <optional>
    latexmk
]{lwarp}
...
\xindyindex
\makeindex
\makeindex[secondname]
```

lwarpmk will use **latexmk** to create the indexes automatically.

For multiple indexes with memoir and xindy, *without* latexmk:

lwarpmk must be told how to create the indexes:

```
\documentclass{memoir}
\usepackage[
  xindy,
  PrintIndexCmd={
    xindy -M lwarp.xdy -L english -C utf8
      <projectname>.idx ;
    xindy -M lwarp.xdy -L english -C utf8
      secondname.idx
  },
  HTMLIndexCmd={
    xindy -M lwarp.xdy -L english -C utf8
      <projectname>_html.idx ;
    xindy -M lwarp.xdy -L english -C utf8
      secondname_html.idx
  }
]{lwarp}
...
\xindyindex
\makeindex
\makeindex[secondname]
```

⚠ WINDOWS *For WINDOWS, replace the four ";" characters with "&".*

<projectname> is the \jobname: if compiling "name.tex", use the filenames name.idx and name_html.idx.

Use

Enter ⇒ `lwarpmk printindex`

Enter ⇒ `lwarpmk htmlindex`

to compile the indexes.

9.5.17 Using a custom makeindex style file

Prog makeindex
File lwarp.ist

When using **makeindex**, **lwarpmk** uses the file lwarp.ist to process the index. This file is over-written by **lwarp** whenever a print version of the document is processed.

To use a custom **makeindex** style file:

1. Copy lwarp.ist to a new filename such as projectname.ist
2. Make changes to projectname.ist. Keep the lines which refer to \hyperindexref. These lines creates the hyperlinks for the HTML index. During print output \hyperindexref becomes a null function.

Opt makeindexStyle

3. In the document source use the makeindexStyle option for **lwarp**:

```
\usepackage[
    ... other options ...
    \textred{makeindexStyle=projectname.ist},
]{lwarp}
```

   Likewise, refer to the custom style file if using \PrintIndexCmd, \HTMLIndexCmd, or \LatexmkIndexCmd.

4. Recompile the print version, which causes **lwarp** to rewrite the lwarpmk.conf configuration file. This tells **lwarpmk** to use the custom projectname.ist file instead of lwarp.ist.

9.5.18 Using a custom xindy style file

Prog xindy
File lwarp.xdy

When using **xindy**, **lwarpmk** uses the file lwarp.xdy to process the index. This file is over-written by **lwarp** whenever a print version of the document is processed.

To use a custom **xindy** style file:

1. Copy lwarp.xdy to a new filename such as projectname.xdy
2. Make changes to projectname.xdy.

   Keep the lines which refer to \hyperindexref:

```
(define-attributes (("hyperindexref")))
(markup-locref :open "\hyperindexref{" :close "}")
...
(markup-locref :open "\textit{\hyperindexref{" :close "}}" :attr "textit")
```

   These lines create the hyperlinks for the HTML index. During print output \hyperindexref becomes a null function.

   To create custom styles, refer to the lines for \textbf and \textit.

Opt xindyStyle

3. In the document source use the xindyStyle option for **lwarp**:

```
\usepackage[
    ... other options ...
    \textred{xindyStyle=projectname.xdy},
]{lwarp}
```

   Likewise, refer to the custom style file if using \PrintIndexCmd, \HTMLIndexCmd, or \LatexmkIndexCmd.

4. Recompile the print version, which causes **lwarp** to rewrite the lwarpmk.conf configuration file. This tells **lwarpmk** to use the custom projectname.xdy file instead of lwarp.xdy.

9.5.19 Additional indexing limitations

xindy with **hyperref**

xindy and **hyperref** may not work well together for print output with "see", "see also", reference ranges, or stylized index references. It may be necessary to turn off hyper-referencing for indexes:

```
\usepackage[hyperindex=false]{hyperref}
```

⚠ empty index

If an HTML index is empty, it may be necessary to add the following before **lwarp** is loaded:

```
\usepackage{morewrites}
\morewritessetup{allocate=10}
...
\usepackage{lwarp}
```

makeindex custom display styles

When using **makeindex**, custom display styles are possible:

```
\begin{warpprint}
\newcommand{\notesstyle}[1]{#1nn}
\end{warpprint}

\begin{warpHTML}
\makeatletter
\newcommand{\notesstyle}[1]{\LWR@doindexentry{#1} notes }
\makeatother
\end{warpHTML}
...
A sentence.\index{key|notesstyle}
```

xindy custom display styles

For custom styles with **xindy**, see `lwarp.xdy` for `\textbf` and `\textit` as examples.

9.5.20 Index positions, TOC, tocbibind

placement and TOC options

An index may be placed inline with other HTML text, or on its own HTML page:

Pkg makeidx

Inline, with a manual TOC entry:

A commonly-used method to introduce an index in a LaTeX document:

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex
```

Pkg makeidx **On its own HTML page, with a manual TOC entry:**

```
\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex
```

Pkg tocbibind **Inline, with an automatic TOC entry:**

The **tocbibind** package may be used to automatically place an entry in the TOC.

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex
```

Pkg tocbibind **On its own HTML page, with an automatic TOC entry:**

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex
```

Opt tocbibind numindex
numbered index section

Use the **tocbibind** `numindex` option to generate a numbered index. Without this option, the index heading has no number.

Other packages, such as **imakeidx**, may also have options for including the index in the Table of Contents.

Pkg tocloft
⚠ **tocloft** & other packages

If using **tocloft** with **tocbibind**, **anonchap**, **fncychap**, or other packages which change chapter title formatting, load **tocloft** with its `titles` option, which tells **tocloft** to use standard LaTeX commands to create the titles, allowing other packages to work with it.

9.6 Math

9.6.1 Rendering tradeoffs

Math rendering

Math may be rendered as SVG graphics or using the MathJax JavaScript display engine.

SVG files

Rendering math as images creates a new SVG file for each expression, except that an MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to SVG only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change.

SVG inline

The SVG images are currently stored separately, but they could be encoded in-line directly into the HTML document. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files

Others LaTeX-to-HTML converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML

Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 10 regarding EPUB output with MathJax.

9.6.2 SVG option

SVG math option

For SVG math, math is rendered as usual by LaTeX into the initial PDF file using the current font[14], then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the LaTeX code which generated the math, allowing copy/paste of the LaTeX math expression into other documents.

SVG image font size

For the `lateximage` environment, the size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, which defaults to:

```
\renewcommand{\LateximageFontSizeName}{normalsize}
```

For inline SVG math, font size is instead controlled by `\LateximageFontScale`, which defaults to:

```
\newcommand*{\LateximageFontScale}{.75}
```

SVG math copy/paste

For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the LaTeX code for the math expression. For $\mathcal{AMS}$ environments with multiple numbers in the same environment, only the first and

[14] See section 411 regarding fonts and fractions.

last is copy/pasted, as a range. No tags are listed inside a starred AMS environment, although the \tag macro will still appear inside the LaTeX math expression.

⚠ SVG math in TeX boxes

SVG math does not work inside TeX boxes, since a \newpage is required before and after each image.

9.6.3 MathJax option

MathJax math option

Prog MathJax

The popular MathJax alternative (mathjax.org) may be used to display math.

When MathJax is enabled, math is rendered twice:

1. As regular LaTeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of LaTeX, and
2. As detokenized printed LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current LaTeX values, and the MathJax cross-referencing system is ignored in favor of the LaTeX internal system, seamlessly integrating with the rest of the LaTeX code.

9.6.4 Customizing MathJax

MathJax does not have preexisting support every possible math function. Additional MathJax function definitions may be defined. These will be declared at the start of each HTML page, and thus will have a global effect.

Examples:

```
\CustomizeMathJax{
    \newcommand{\expval}[1]{\langle#1\rangle}
    \newcommand{\abs}[1]{\lvert#1\rvert}
}
\CustomizeMathJax{\newcommand{\arsinh}{\text{arsinh}}}
\CustomizeMathJax{\newcommand{\arcosh}{\text{arcosh}}}
\CustomizeMathJax{\newcommand{\NN}{\mathbb{N}}}
```

9.6.5 MathJax limitations

MathJax limitations

Prog MathJax

Limitations when using MathJax include:

chapter numbers

- In document classes which have chapters, \tagged equations have the chapter

number prepended in HTML output, unlike LaTeX. \tag* equations (correctly) do not. This may be improved with future versions of the MathJax support script.

https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY

subequations

- MathJax itself does not support subequations. This may be improved by parsing the LaTeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

- Footnotes inside equations are not yet supported while using MathJax.

lateximage

- Math appearing inside a lateximage, and therefore also inside a Ti*k*z or picture environment, is rendered as SVG math even if MathJax is used in the rest of the document.

siunitx

⚠ siunitx inside an equation

- Usage of **siunitx** inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use \SI or \si inside \text, where it will be rendered as normal text.

  https://github.com/burnpanck/MathJax-siunitx

  Also see section 9.6.12.

tabbing

- A tabbing environment is emulated using an HTML <pre>.While MathJax is enabled inside tabbing, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.

⚠ other macros and packages

- Other math-related macros and packages are not supported by MathJax, including \ensuremath, **bigdelim**, **units**, and **nicefrac**, along with occasionally-used macros such as \footnote and \relax.

9.6.6 Catcode changes

preamble macros with math

The math shift character $ is not set for HTML output until after the preamble. Macros defined in the preamble which contain $ must be enclosed between \StartDefiningMath and \StopDefiningMath to temporarily change to the HTML meaning of $:

```
\StartDefiningMath
\newcommand{...}
\StopDefiningMath
```

As an alternative, use \(and \) instead of $, in which case \StartDefiningMath and \StopDefiningMath are not necessary.

If a package defines macros using $, it may be nessary to use \StartDefiningMath and \StopDefiningMath before and after loading the package.

9.6.7 Dynamic math

inline dynamic math

An inline math expression is usually converted to a reusable hashed SVG math image, or a MathJax expression. The hash or expression depends on the contents of the math expression. In most cases this math expression is static, such as `$x+1$`, so the image can be reused for multiples instances of the same expression. In some cases, the math expression includes a counter or other object which may change between uses. The macro `\StartDynamicMath` may be used before a dynamic math expression, and `\StopDynamicMath` after. Doing so tells **lwarp** to use an unhashed SVG math image, even if MathJax is in use. See section 41.

9.6.8 Display math

`\displaymathnormal`

By default, or when selecting `\displaymathnormal`, math display environments print their contents in MATHJAX, and render their contents in SVG math as well as use their contents in the `alt` tag of HTML output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated Ti*k*z pictures, compilation will fail.

`\displaymathother`

When selecting `\displaymathother`, it is assumed that the contents are more complicated than "pure" math. An example is an elaborate Ti*k*z picture, which will not render in MATHJAX and will not make sense as an HTML `alt` tag. In this mode, MATHJAX is turned off, math display environments become SVG images, even if MATHJAX is selected, and the HTML `alt` tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as Ti*k*z pictures are more likely to compile successfully.

9.6.9 chemformula package

⚠ **chemformula** with MATHJAX

chemformula works best without MATHJAX. If MATHJAX is used, `\displaymathother` must be used before `array`, and then `\displaymathnormal` may be used after. (The **chemformula** package adapts to `array`, but does not know about MATHJAX, and MATHJAX does not know about **chemformula**.)

While using MATHJAX, `\displaymathother` may also be used for other forms of display and inline math which contain **chemformula** expressions.

9.6.10 mhchem package

See section 276.

9.6.11 ntheorem package

Pkg ntheorem

⚠ Font control

This conversion is not total. Font control is via css, and the custom LaTeX font settings are ignored.

⚠ Equation numbering

ntheorem has a bug with equation numbering in $\mathcal{AMS}$ environments when the option `thref` is used. **lwarp** does not share this bug, so equations with `\split`, etc, are numbered correctly with **lwarp**'s HTML output, but not with the print output. It is recommended to use **cleveref** instead of **ntheorem**'s `thref` option.

9.6.12 siunitx package

Pkg siunitx

fractions

Due to **pdftolatex** limitations, fraction output is replaced by symbol output for `per-mode` and `quotient-mode`.

⚠ math mode required

Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the **siunitx** extension for MathJax is not currently hosted at any public CDN, thus **siunitx** is not usable with MathJax unless a local copy of this extension is created first.

⚠ tabular

Tabular `S` columns are rendered as simple `c` columns, and tabular `s` columns are not supported. These may be replaced by `c` columns with each cell contained in `\num` or `\si`.

9.6.13 units and nicefrac packages

Pkg units

Pkg nicefrac

units and **nicefrac** work with **lwarp**, but MathJax does not have an extension for **units** or **nicefrac**. These packages do work with **lwarp**'s option `svgmath`.

9.6.14 newtxmath package

Pkg newtxmath

⚠ loading sequence

The proper load order is:

```
...
\usepackage{lwarp}
...
\usepackage{amsthm}
\usepackage{newtxmath}
...
```

9.7 Graphics

Pkg graphics
Pkg graphicx
\includegraphics file formats

For \includegraphics with .pdf or .eps files, the user must provide a .pdf or .eps image file for use in print mode, and also a .svg, .png, or .jpg version of the same image for use in HTML.

```
\includegraphics{filename} % print:.pdf/.eps HTML:.svg, etc.
```

For print output, **lwarp** will automatically choose the .pdf or .eps format if available, or some other format otherwise. For HTML, one of the other formats is used instead.

If a .pdf or .eps image is referred to with its file extension, the extension will be changed to .svg for HTML:

```
\includegraphics{filename.pdf} % uses .svg in HTML
\includegraphics{filename.eps} % uses .svg in HTML
```

Prog pdftocairo
PDF to SVG

To convert a PDF image to SVG, use the utility **pdftocairo**:

Enter ⇒ `pdftocairo -svg filename.pdf`

Prog lwarpmk pdftosvg

For a large number of images, use **lwarpmk**:

Enter ⇒ `lwarpmk pdftosvg *.pdf` (or a list of filenames)

Prog lwarpmk epstopdf
Prog epstopdf
epstopdf package

For EPS images converted to PDF using the package **epstopdf**, use

Enter ⇒ `lwarpmk pdftosvg *.PDF`

to convert to SVG images.

DVI latex

When using DVI **latex**, it is necessary to convert EPS to PDF and then to SVG:

Enter ⇒ `lwarpmk epstopdf *.eps` (or a list of filenames)

Enter ⇒ `lwarpmk pdftosvg *.pdf` (or a list of filenames)

PNG and JPG

For PNG or JPG while using **pdflatex**, **lualatex**, or **xelatex**, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

GIF

GIF files may be used for HTML, but another format must also be provided for print output.

file extension priorities

If a file extension is not used, for HTML the file extension priorities are: SVG, GIF, PNG, then JPG.

⚠ **graphics** vs. **graphicx**

If using the older **graphics** syntax, use both optional arguments for \includegraphics.

⚠ viewports

A single optional parameter is interpreted as the newer **graphicx** syntax. Note that viewports are not supported by **lwarp** — the entire image will be shown.

units

For `\includegraphics`, avoid `px` and % units for width and height, or enclose them inside `warpHTML` environments. For font-proportional image sizes, use `ex` or `em`. For fixed-sized images, use `cm`, `mm`, `in`, `pt`, or `pc`. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area. Do not use the `scale` option, since it is not well supported by HTML browsers.

options

`\includegraphics` accepts `width` and `height`, `origin`, `rotate` and `scale`, plus a new `class` key.

HTML class

With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

`\rotatebox`

`\rotatebox` accepts the optional `origin` key.

⚠ browser support

`\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The css3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike LaTeX, so expect some ugly results for scaling and rotating.

9.7.1 tikz package

Pkg `tikz`

⚠ displaymath and matrices

If using display math with `tikzpicture` or `\tikz`, along with matrices with the `&` character, the document must be modified as follows:

```
\usepackage{tikz}
\tikzset{every picture/.style={ampersand replacement=\&}}
```

and each instance of `&` in the **tikz** expression must be replaced with `\&`.

9.7.2 grffile package

Pkg `grffile`

⚠ matching PDF and SVG

grffile is supported as-is. File types known to the browser are displayed, and unknown file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

9.7.3 color package

Pkg `color`

color is superceded by **xcolor**, and **lwarp** requires several of the features of **xcolor**.

⚠ missing colors

It should be sufficient for the user's document to load **color** then load **xcolor** as well.

9.7.4 xcolor package

Pkg xcolor

\colorboxBlock and \fcolorboxBlock

`\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a `<span>`. These `<div>`s are set to `display: inline-block` so adjacent `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by **lwarp**'s core if **xcolor** is loaded.

background: none

`\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case only the frame is drawn, which can be useful for HTML.

color support

Color definitions, models, and mixing are fully supported without any changes required.

colored tables

`\rowcolors` is supported, except that the optional argument is ignored so far.

colored text and boxes

`\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

\color and \pagecolor

`\color` and `\pagecolor` are ignored. Use CSS or `\textcolor` where possible.

9.7.5 epstopdf package

Pkg epstopdf

⚠ convert to .svg

Images with an `.eps` extension will be converted to `.pdf`. The HTML output uses the `.svg` version, so use

Enter ⇒ `lwarpmk pdftosvg <listofPDFfiles>`

to generate `.svg` versions.

9.7.6 pstricks package

Pkg pstricks

⚠ use pspicture

All **pstricks** content should be contained inside a `pspicture` environment.

9.7.7 pdftricks package

Pkg pdftricks

⚠ convert image files

The **pdftricks** image files `<jobname>-fig*.pdf` must be converted to `.svg`, or else a

missing file error will occur. The image files must also be converted again whenever they change. To convert the images:

Enter ⇒ `lwarpmk pdftosvg <jobname>-fig*.pdf`

9.7.8 overpic package

Pkg `overpic`

⚠ scaling

The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

9.8 Tabbing

The `tabbing` environment works, except that SVG math and `lateximages` do not yet work inside the environment.

math in `tabbing`

If math is used inside `tabbing`, place `tabbing` inside a `lateximage` environment, which will render the entire environment as a single SVG image.

9.9 Tabular

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, * column specifiers, **siunitx** S columns, or the packages **multirow**, **longtable**, **supertabular**, or **xtab**.

Defining environments:

⚠ misplaced alignment
alignment tab character &

- When defining environments or macros which include `tabular` and instances of the `&` character, it may be necessary to make `&` active before the environment or macro is defined, then restore `&` to its default catcode after, using the following commands. These are are ignored in print mode.

```
\StartDefiningTabulars
<define macros or environments using tabular and &
here>
\StopDefiningTabulars
```

⚠ floatrow

This includes before and after defining any macro which used `\ttabbox` from **floatrow**.

⚠ `tabular` inside another environment

- When creating a new environment which contains a `tabular` environment, **lwarp**'s emulation of the tabular does not automatically resume

when the containing enviroment ends, resulting in corrupted HTML rows. To fix this, use \ResumeTabular as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\StopDefiningTabulars
```

Cell contents:

⚠ paragraphs

- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.

⚠ \multirow

- For **multirow**, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

```
... &  \multirow{2}{.5in}{text}  & ...
... &  \mrowcell                 & ...
```

vposn

Note that recent versions of **multirow** include a new optional vposn argument.

- The **multirow** documentation regarding colored cells recommends using a negative number of rows. This will not work with **lwarp**, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.
- See section 285.2 for \multicolumnrow.

⚠ \multicolumn & \multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
... &  \multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text}  & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
```

vposn

Note that recent versions of **multirow** include a new optional `vposn` argument.

⚠ macro in a table
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \\
```

Column specifiers:

⚠ * column specification

- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of `@` and `!` is used at each column, and they are used in that order.

\multirow

- In `\multirow` cells, the print version may have extra instances of `<`, `>`, `@`, and `!` cells on the second and later rows in the `\multirow` which do not appear in the HTML version.

⚠ \newcolumntype

- `\newcolumntype` is ignored; unknown column types are set to `l`.

Rules:

- Doubled `\hlines`, `\midrules`, and vertical rules are supported.

vertical rules

- Vertical rules next to either side of an `@` or `!` column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with `@` or `!` columns, and full-width rules ignore trim.

full-width rules

- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use `\cmidrule` followed by `\bottomrule`, it may be necessary to use:

```
\cmidrule{2-3} \\[-2ex]
\bottomrule
```

The optional `-2ex` is ignored in HTML but improves the visual formatting in the print output.

⚠ \warpprintonly
misplaced \noalign

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warpHTML` environment, if a "misplaced `\noalign`" error occurs, change

```
This & That \endhead
```

to

```
\warpprintonly{This & That \endhead}
```

and likewise with the other \end headings. Keep the \endfirsthead row unchanged, as it is still relevent to HTML output.

colortbl:

⚠ row/cell color

Only use \rowcolor and \cellcolor at the start of a row, in that order.

colortbl ignores the overhang arguments.

Other:

- **tabularx** ignores the width, but X columns do produce paragraph columns or multicolumns.

longtable headings

- For **longtable**, place headings and footings which do not apply to HTML inside \warpprintonly{}.

⚠ S columns

- For S columns (from the **siunitx** package), while producing print output, anything non-numeric must be placed inside { } braces, including commands such as \multirow. While producing HTML output, though, anything placed inside braces is not seen by **lwarp**'s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

9.9.1 longtable package

Pkg longtable ⚠ Longtable \endhead, \endfoot, and \endlastfoot rows are not used for HTML, and these rows should be disabled. Use

```
\warpprintonly{row contents}
```

instead of

```
\begin{warpprint} ... \end{warpprint}
```

Doing so helps avoid "Misplaced \noalign." when using \begin{warpprint}.

Keep the \endfirsthead row, which is still relevent to HTML output.

⚠ \kill is ignored, place a \kill line inside

```
\begin{warpprint} ... \end{warpprint}
```

or place it inside \warpingprintonly.

⚠ lateximage **longtable** is not supported inside a lateximage.

9.9.2 supertabular and xtab packages

Pkg supertabular
Pkg xtab
⚠ misplaced alignment
alignment tab character &

For \tablefirsthead, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\StopDefiningTabulars
```

See section 9.9.

⚠ lateximage

supertabular and **xtab** are not supported inside a lateximage.

9.9.3 bigdelim package

Pkg bigdelim
⚠ use \mrowcell

\ldelim and \rdelim use \multirow, so \mrowcell must be used in the proper number of empty cells in the same column below \ldelim or \rdelim, but not in cells which are above or below the delimiter:

```
\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\{}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

<> a b
left { c d
e f
<> g h

9.10 Floats

9.10.1 Float contents alignment

figure & table alignment

\centering, etc. are honored in a figure or table if they are the first command inside the float:

```
\begin{table*}
\centering
\caption{A Table}
...
```

9.10.2 float, trivfloat, and/or algorithmicx together

Pkg float
Pkg trivfloat
Pkg algorithmx

If using \newfloat, **trivfloat**, and/or **algorithmicx** together, see section 390.1.

9.10.3 caption and subcaption packages

⚠ package conflicts

Pkg caption
Pkg subcaption

To pass options to caption, select the options before loading **lwarp**:

⚠ options

```
\documentclass{article}
...
\PassOptionsToPackage{options_list}{caption}
...
\usepackage{lwarp}
...
\usepackage{caption}
```

To ensure proper float numbering, set caption positions such as:

```
\captionsetup[table]{position=top}
\captionsetup[figure]{position=bottom}
```

Similarly for **subtable**, **subfigure**, and **longtable**.

9.10.4 subfig package

Pkg subfig

⚠ lof/lotdepth

At present, the package options for lofdepth and lotdepth are not working. These counters must be set separately after the package has been loaded.

horizontal spacing

In the document source, use \hfill and \hspace* between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

9.10.5 floatrow package

Pkg floatrow

⚠ misplaced alignment
alignment tab character &

Use \StartDefiningTabulars and \StopDefiningTabulars before and after defining macros using \ttabbox with a tabular inside. See section 9.9.

⚠ **subfig** package

When combined with the **subfig** package, while inside a subfloatrow \ffigbox and \ttabbox must have the caption in the first of the two of the mandatory arguments.

⚠ \FBwidth, \FBheight

The emulation of **floatrow** does not support \FBwidth or \FBheight. These values are pre-set to .3\linewidth and 2in. Possible solutions include:

- Use fixed lengths. **lwarp** will scale the HTML lengths appropriately.
- Use warpprint and warpHTML environments to select appropriate values for each case.
- Inside a warpHTML environment, manually change \FBwidth or \FBheight before the \ffigbox or \ttabbox. Use \FBwidth or \FBheight normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

9.10.6 keyfloat package

Pkg keyfloat
⚠ keywrap

If placing a \keyfig[H] inside a keywrap, use an absolute width for \keyfig, instead of lw-proportional widths. (The [H] option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the lw option.)

9.11 Koma-Script

Cls komascript

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

\titlehead, \subject, \captionformat, \figureformat, and \tableformat are not yet emulated.

⚠ Not fully tested! Please send bug reports!

Some features have not yet been tested. Please contact the author with any bug reports.

9.12 Memoir

Cls memoir
⚠ options clash

While emulating **memoir**, **lwarp** pre-loads a number of packages (section 422.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading **lwarp**:

```
\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}
```

`\verbfootnote` is not supported.

`\newfootnoteseries`, etc. are not supported.

lwarp loads **pagenote** to perform **memoir**'s pagenote functions, but there are minor differences in `\pagenotesubhead` and related macros.

Poem numbering is not supported.

The `verbatim` environment does not yet support the **memoir** enhancements. It is currently recommended to load and use **fancyvrb** instead.

The **memoir** glossary system is not yet supported by **lwarpmk**. The **glossaries** package may be used instead, but does require the glossary entries be changed from the **memoir** syntax to the **glossaries** syntax.

9.13 Miscellaneous

9.13.1 verse and memoir

Pkg `verse`

When using **verse** or **memoir**, always place a `\\` after each line.

Cls `memoir`
`\attrib`

The documentation for the **verse** and **memoir** packages suggest defining an `\attrib` command, which may already exist in current documents, but it will only work for print output. **lwarp** provides `\attribution`, which works for both print and HTML output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

Len `\vleftskip`
Len `\vleftmargini`
Len `\HTMLvleftskip`
Len `\HTMLleftmargini`

These lengths are used by **verse** and **memoir** to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any `verse` environment, and persist until they

are manually changed again. One reason to change \HTMLleftmargini is if there is a wide \flagverse in use, such as the word “Chorus”, in which case the value of \HTMLleftmargini should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

⚠ spacing

Horizontal spacing relies on **pdftotext**’s ability to discern the layout (-layout option) of the text in the HTML-tagged PDF output. For some settings of \HTMLleftmargini or \HTMLleftskip the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

9.13.2 newclude package

Pkg newclude

newclude modifies \label in a non-adaptive way, so **newclude** must be loaded before **lwarp** is loaded:

⚠ loading

```
\documentclass{article}
...<font setup>
\usepackage{newclude}
\usepackage[warpHTML]{lwarp}
...
```

9.13.3 babel package

Pkg babel

\CaptionSeparator

When French is used, the caption separator is changed to a dash. The following may be used to restore it to a colon:

```
\renewcommand*{\CaptionSeparator}{:~}
```

punctuation spaces

Also when French is used, **lwarp** creates fixed-width space around punctuation by patching \FBcolonspace, \FBthinspace, \FBguillspace, \FBmedkern, \FBthickkern, \FBtextellipsis, and the tilde. If the user’s document also changes these parameters, the user’s changes should be placed inside a warpprint environment so that the user’s changes do not affect the HTML output.

⚠ customized spacing

9.13.4 polyglossia package

Pkg polyglossia

lwarp uses **cleveref**, which has some limitations when using **polyglossia**, possibly resulting in the error

```
! Undefined control sequence.  ... \@begindocumenthook
```

To test compatibility, add

```
\usepackage{cleveref}
```

near the end of the preamble (as the last package to be loaded), and try to compile the print version. It may be necessary to set

```
\setdefaultlanguage{english}
```

or some other language supported by **cleveref**, then select other languages using `\setotherlanguages`.

Once the print version works with **cleveref** and **polyglossia**, the HTML version should work as well using **lwarp**.

9.13.5 todonotes and luatodonotes packages

Pkg `todonotes`
Pkg `luatodonotes`

The documentation for **todonotes** and **luatodonotes** have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

9.13.6 fixme

Pkg `fixme`

External layouts (`\fxloadlayouts`) are not supported.

⚠ external layouts

User control is provided for setting the HTML styling of the "faces". The defaults are as follows, and may be changed in the preamble after **fixme** is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

9.13.7 xparse

Pkg `xparse`

To remove from the log any warnings about redeclaring objects, place the following before **lwarp** is loaded:

```
\usepackage[log-declarations=false]{xparse}
```

10 EPUB conversion

lwarp does not produce EPUB documents, but it may be told to modify its HTML output to greatly assist in the conversion. An external program may then be used to finish the conversion to EPUB.

<meta> author

To assign the author's name for regular **lwarp** HTML files, and also for the EPUB, use `\HTMLAuthor {⟨name⟩}`. This assigns the name to the `<meta>` author element. It may be set empty, and it defaults to `\theauthor`.

A special boolean is provided to simplify the process of converting **lwarp** HTML output to EPUB:

FormatEPUB

Bool FormatEPUB
Default: false

FormatEPUB changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.

To help convert **lwarp** HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

to the project's source preamble after `\usepackage{lwarp}`. The EPUB version of the document cannot co-exist with the regular HTML version, so

```
Enter ⇒  lwarpmk cleanall
Enter ⇒  lwarpmk html
Enter ⇒  lwarpmk limages
```

to recompile with the FormatEPUB boolean turned on. Several changes are then made to the HTML output:

- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.

Calibre

The resulting files will be ready to be loaded into an EPUB conversion program, such as the open-source program **Calibre** (https://calibre-ebook.com/).

⚠ search order

The EPUB conversion program must know what order the files are included. For **lwarp** projects, set the EPUB conversion software to do a breadth-first search of the files. For **Calibre**, this option is found in

Preferences → Plugins → File type plugins → HTML to Zip

⚠ encoding

Check the box Add linked files in breadth first order. Set the document encoding as `utf-8`, which is what **lwarp** generates for HTML, even if the original printed document uses some other encoding.

⚠ section breaks

The EPUB-conversion program must also know where the section breaks are located. For a list of **lwarp**'s section headings, see table 7. For example, an `article` class document would break at `\section`, which is mapped to HTML heading level `<h4>`, whereas a `book` class document would break at `\chapter`, which is HTML heading level `<h3>`. For **Calibre**, this option is found in

> Preferences → Conversion (Common Options) → Structure Detection → Detect chapters at (XPath expression)

Select the "magic wand" to the right of this entry box, and set the first entry

> Match HTML tags with tag name:

to "h4". (Or "h3" for document classes with `\chapters`.) The Detect chapters at field should then show

> //h:h4 — or — //h:h3

This option is also available on the main tool bar at the Convert books button.

Once these settings have been made, the **lwarp**-generated HTML files may be loaded by **Calibre**, and then converted to an EPUB.

MathJax support

MathJax may be used in EPUB documents. Some e-readers include MathJax, but any given reader may or may not have a recent version, and may or may not include extensions such as support for **siunitx**.

lwarp adds some modifications to MathML to support equations numbered by chapter. These modifications may not be compatible with the e-reader's version of MathJax, so **lwarp** requests that a known version be loaded instead. In some cases chapter numbering of equations still doesn't work.

Until math support in EPUB documents is improved, it is recommended to use SVG images instead of MathJax, especially for equations numbered by chapter, or where **siunitx** support is important.

11 Word-processor conversion

lwarp may be told to modify its HTML output to make it easier to import the HTML document into a word processor. At the time of this writing, it seems that LIBREOFFICE works best at preserving table layout, but it still has some limitations, such as an inability to automatically assign figure and table frames and captions according to user-selected HTML classes. **lwarp** provides some assistance in locating these frame boundaries, as shown below.

11.1 Activating word-processor conversion

A special boolean is provided to simplify the process of converting **lwarp** HTML output to EPUB:

FormatWP

Bool `FormatWP`
Default: `false`

Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Additionally, honors the booleans `WPMarkFloats`, `WPMarkMinipages`, `WPMarkTOC`, and `WPMarkLOFT`.

To help modify **lwarp** HTML output for easier import to a word processor, add

```
\booltrue{FormatWP}
```

to the project's source preamble after **lwarp** is loaded. The following changes are formatting adjustments then made to the HTML output:

- If using a class without chapters, `\section` and lower are shifted up in level for the HTML heading tags. The CSS has not been changed, so the section heading formats will not match the normal HTML output, but when imported to **LibreOffice Writer** the higher section headings will import as **Heading 1** for the title, **Heading 2** for `\section`, etc.
- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.
- Forces single-file output.
- Turns off HTML debugging comments. These are comments appearing inside the HTML code, marking the opening/closing of sections and `<div>`s, but they are no longer useful when the document has been imported into a word processor.

- An additional `<div>` with an `id` encapsulates each float and minipage, which on import into **LibreOffice Writer** causes a thin frame to appear around the text block for each.
- Float captions are given an explicit italic formatting.
- Tabular rule borders are made explicit for **LibreOffice Writer**. LibreOffice displays a light border around each cell while editing, even those which have no border when printed, and **lwarp** also uses a light border for thin rules, so it will be best to judge the results using the print preview instead of while editing in LibreOffice.
- `\includegraphics` and SVG math width and height are made explicit for LibreOffice.
- `\hspace` is approximated by a number of `\quad`s, and rules are approximated by a number of underscores.
- Explicit HTML styles are given to:
  - `\textsc`, etc.
  - `\underline`, **soul** and **ulem** markup.
  - `center`, `flushleft`, `flushright`.
  - `\marginpar`, **keyfloat**, **sidenotes**, **floatflt**, and **wrapfig**.
  - **fancybox** `\shadowbox`, etc.
  - The LaTeX and TeX logos.
- Honors several booleans:

  `WPMarkFloats`: Marks the begin and end of floats.

  `WPMarkMinipages`: Marks the begin and end of minipages.

  `WPMarkTOC`: Marks the location of the Table of Contents.

  `WPMarkLOFT`: Marks the locations of the List of Figures/Tables.

  `WPMarkMath`: Prints LaTeX math instead of using images.

  `WPTitleHeading`: Adjusts title and section headings.

  Several of these may be used to add markers to the HTML text which help determine where to adjust the word processor document after import.

11.2 Additional modifications

WPMarkFloats

Adds

```
=== begin table ===
...
=== end ===
```

or

```
=== begin figure ===
...
=== end ===
```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions.

Bool `WPMarkFloats`
Default: `false`

WPMarkMinipages

Adds

```
=== begin minipage ===
...
=== end minipage ===
```

around minipages while formatting for word processors. This helps identify boundaries of minipages to be manually converted to word-processor frames.

Bool `WPMarkMinipages`
Default: `false`

WPMarkTOC

While formatting for word processors, adds

```
=== table of contents ===
```

where the Table of Contents would have been. This helps identify where to insert the actual TOC.

If set `false`, the actual TOC is printed instead.

Bool `WPMarkTOC`
Default: `true`

WPMarkLOFT

While formatting for word processors, adds

```
=== list of figures ===
=== list of tables ===
```

and/or

Bool WPMarkLOFT
Default: false

where each of these lists would have been. This helps identify where to insert the actual lists.

If set `false`, the actual lists are printed instead.

WPMarkMath

While formatting for word processors, prints math as LaTeX code instead of creating SVG images or MathJax. This is useful for cut/paste into the **LibreOffice Writer TeXMaths** extension.

Bool WPMarkMath
Default: false
Prog TeXMaths
siunitx

When using the **siunitx** package, enter

```
\usepackage{siunitx}
```

in the **TeXMaths** preamble. Equation numbering is problematic for AMS math environments.

WPTitleHeading

While formatting for word processors, `true` sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to `false` to cause the title to be plain text, and the section headings to begin at **Heading 1**.

Bool WPTitleHeading
Default: false
section headings

See table 6 on page 169.

11.3 Recommendations

TOC, LOF, LOT

For use with **LibreOffice Writer**, it is recommended to:

1. Set \booltrue{FormatWP}.
2. Set \booltrue{WPMarkTOC} and \boolfalse{WPMarkLOFT}.
3. Use **lwarp** to generate the HTML document.
4. Copy/paste from the HTML document into an empty **LibreOffice Writer** document.
5. Manually insert a LibreOffice TOC in the LibreOffice document.

Table 6: Section HTML headings for word-processor conversion

	HTML headings*			
	With \chapter		Without \chapter	
	WPTitleHeading		WPTitleHeading	
Section	true	false	true	false
Title	<h1>	plain	<h1>	plain
\part	<h2>	<h1>	<h2>	<h1>
\chapter	<h3>	<h2>	—	—
\section	<h4>	<h3>	<h3>	<h2>
\subsection	<h5>	<h4>	<h4>	<h3>
\paragraph	<h6>	<h5>	<h5>	<h4>
\subparagraph	span	<h6>	<h6>	<h5>

* For default depths when not FormatWP, see table 7 on page 182.

6. Manually add frames around each float, adding a caption which is cut/pasted from each float's simulated caption.
7. Manually create cross references.

This process yields a document with an actual LibreOffice Table of Contents, but a simulated List of Figures and List of Tables.

siunitx

For **siunitx**, remember to adjust the preamble as mentioned above.

LO view border options

LibreOffice has options in the **View** menu to turn on/off the display of thin borders around table cells and text objects.

11.4 Limitations

Floats and captions are not explicitly converted to LibreOffice floats with their own captions. Floats are surrounded by a thin frame in the LibreOffice editor, and may be marked with WPMarkFloats, but are not given a proper LibreOffice object frame. Captions are given an explicit italic formatting, but not a proper LibreOffice paragraph style.

Cross references are not actual LibreOffice linked cross references.

The List of Figures and List of Tables are not linked. The pasted pseudo LOF and LOT match the numbering of the LaTeX and HTML versions.

Equation numbering is not automatic, but the equation numbers in SVG math will match the LaTeX and HTML output. SVG math is recommended when using the $\mathcal{AMS}$ environments, which may have multiple numbered equations per object.

As of when last checked, LibreOffice ignores the following:

- Minipage alignment.
- Tabular cell vertical alignment.
- Image rotation and scaling.
- Rounded border corners, which are also used by:
  - `\textcircled`
  - **booktabs** trim
- `\hspace` and rules, also used by **algorithmic**.
- Coloring of text decorations, used by **soul** and **ulem**.
- Overline text decoration, used by **romanbar**.

Libreoffice also has limitations with frames and backgrounds:

- Multiple lines in an object are framed individually instead of as a whole.
- Nested frames are not handled correctly.
- Images inside boxes are not framed correctly.
- Spans with background colors and frames are not displayed correctly.

12 Modifying lwarp

locating something

To quickly find the source for a package in `lwarp.dtx`, search for `*packagename`, such as `*siunitx`.

Likewise, to quickly find the source for a file in `lwarp.dtx`, search for `*filename`, such as `*lwarp.css`.

Purely text-based packages probably will work as-is when generating HTML.

Look to existing code for ideas on how to expand into new code.

image of TEX output

An environment may be converted to a `lateximage` then displayed with an image of the resulting LaTeX output. See section 84 for an example of the `picture` environment.

CSS classes

To create a custom HTML block or inline CSS class, see section 49.8.

print/HTML macros

To create print and HTML versions of the same macro or environment, see section 33.

⚠ TEX boxes

Any TeX boxes must be undone, as SVG math or `lateximages` require `\newpage`, which will not work in a TeX box.

index recreation

To recreate the index for the **lwarp** documentation:

```
makeindex -s gglo.ist -o lwarp.gls lwarp.glo
makeindex -s gind.ist lwarp.idx
```

12.1 Creating a development system

The following creates a local development system for **lwarp** on a TeXLive system in a Unix-like environment. Doing so allows anything requesting **lwarp** to use the development version instead of whichever version is installed in TeXLive.

Create a development directory:

Place into this directory `lwarp.dtx` and `lwarp.ins`.

To create `lwarp.sty`, execute

Enter ⇒ `pdflatex lwarp.ins`

which creates `lwarp.sty` and several hundred additional `lwarp-*.sty` files for the various packages which are supported.

To create the documentation `lwarp.pdf`, execute

Enter ⇒ `pdflatex lwarp.dtx`

To make the development files visible to other projects:

Create the directory

```
/usr/local/texlive/texmf-local/tex/latex/local/lwarp
```

Inside this directory, create the file `update`, containing:

```
ln -s /path_to_dev_directory/lwarp*.sty .
ln -s /path_to_dev_directory/lwarp_baseline_marker.png .
ln -s /path_to_dev_directory/lwarp_baseline_marker.eps .
mktexlsr
```

Run ./update now, and whenever a new lwarp-* package is added.

To make the development version of lwarpmk visible to other projects:

```
cd /opt
ln -s /usr/local/texlive/texmf-local/bin/x86_64-linux texbin_local
cd texbin_local
ln -s ../../scripts/lwarp/lwarpmk.lua lwarpmk
cd /usr/local/texlive/texmf-local/scripts/
mkdir lwarp
cd lwarp
ln -s /path_to_dev_directory/lwarpmk.lua lwarpmk
```

Verify that the correct version is found with

Enter ⇒ `which lwarpmk`

To make the local versions visible to the shell:

Paths must be set by the shell startup, such as in `.bashrc` and `.cshrc`:

In `.bashrc`:

```
PATH=/opt/texbin_local:/opt/texbin:$PATH
```

In `.cshrc`:

```
setenv PATH ${HOME}/bin:/opt/texbin_local:/opt/texbin:${PATH}
```

12.2 Modifying a package for lwarp

If a class loads additional packages, it will be required to modify the class for **lwarp**, since **lwarp** must be loaded before most other packages.

To work with **lwarp**, a class must first set up anything which replicates the functions of the basic LaTeX classes, load any required fonts, then load **lwarp**, then finally load and adjust any other required packages.

When creating HTML, **lwarp** redefines the `\usepackage` and `\RequirePackage` macros such that it first looks to see if a `lwarp-<packagename>.sty` version exists. If so, the **lwarp** version is used instead. This modular system allows users to create their own versions of packages for **lwarp** to use for HTML, simply by creating a new package with a `lwarp-` prefix. If placed in the local directory along with the source code, it will be seen by that project alone. If placed alongside the other `lwarp-` packages where TeX can see it, then the user's new package will be seen by any documents using **lwarp**. (Remember `mktexlsr` or `texhash`.)

An `lwarp-<packagename>.sty` package is only used during HTML generation. Its purpose is to pretend to be the original package, while modify anything necessary to create a successful HTML conversion. For many packages it is sufficient to simply provide nullified macros, lengths, counters, etc. for anything which the original package does, while passing the raw text on to be typeset. See the pre-existing `lwarp-` packages for examples.

Anything the user might expect of the original package must be replaced or emulated by the new `lwarp-` package, including package options, user-adjustable counters, lengths, and booleans, and conditional behaviors. In many of these packages, most of the new definitions have a "local" prefix according to the package name, and `@` characters inside the name, which hides these names from the user. In most cases these macros will not need to be emulated for HTML output. Only the "user-facing" macros need to be nullified or emulated.

Each lwarp- package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If "Drop"ped, the original print-version package is ignored, and only the lwarp- version is used. Use this where the original print version is useless for HTML. If "Pass"ed, the original package is loaded first, with the user-supplied options, then the lwarp- version continues loading as well. See section 299 (ntheorem) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable without changes, there is no need to create a lwarp- version.

12.2.1 Adding a package to the lwarp.dtx file

When adding a package to lwarp.dtx for permanent inclusing in **lwarp**, provide the lwarp-<packagename> code in lwarp.dtx, add its entry into lwarp.ins, and also remember to add

```
\LWR@loadafter{<packagename>}
```

to lwarp.dtx in section 25.1. This causes **lwarp** to stop with an error if **packagename** is loaded before **lwarp**. Finally, add an entry in table 1, Supported packages and features, and also the Updates section.

12.3 Modifying a class for lwarp

If a class loads additional packages, it will be required to modify the class for **lwarp**, since **lwarp** must be loaded before most other packages.

To work with **lwarp**, a class must first set up anything which replicates the functions of the basic LaTeX classes, load any required fonts, then load **lwarp**, then finally load and adjust any other required packages.

12.4 Testing lwarp

When changes have been made, test the print output before testing the HTML. The print output compiles faster, and any errors in the printed version will be easier to figure out than the HTML version.

Remember that the configuration files are only rewritten when compiling the printed version of the document.

When changing the source to **lwarpmk** or a CSS file in `lwarp.dtx`:

1. Change the source in `lwarp.dtx`.
2. `pdflatex lwarp.ins`
3. `pdflatex lwarp.dtx`
4. If modifying **lwarpmk** the new version should now be active.
5. If modifying CSS files:
   (a) For the document, `lwarpmk print` to update the CSS files in the project.
   (b) Reload the HTML document to see the effect of the new CSS files.

Sometimes it is worth checking the `<project>_html.pdf` file, which is the PDF containing HTML tags. Also, `<project>_html.html` has the text conversion of these tags, before the file is split into individual HTML files.

It is also worth checking the browser's tools for verifying the correctness of HTML and CSS code.

12.5 Modifying lwarpmk

Prog `lwarpmk`
File `lwarpmk.lua`

In most installations, `lwarpmk.lua` is an executable file located somewhere the operating system knows about, and it is called by typing "`lwarpmk`" into a terminal.

A project-local copy of `lwarpmk.lua` may be generated, modified, and then used to compile documents:

1. Add the `lwarpmk` option to the **lwarp** package.
2. Recompile the printed version of the document. The `lwarpmk` option causes **lwarp** to create a local copy of `lwarpmk.lua`
3. The `lwarpmk` option may now be removed from the **lwarp** package.
4. Copy and rename `lwarpmk.lua` to a new file such as `mymake.lua`.
5. Modify `mymake.lua` as desired.
6. If necessary, make `mymake.lua` executable.
7. Use `mymake.lua` instead of `lwarpmk.lua`.

13 Troubleshooting

13.1 Using the lwarp.sty package

Also see:
Section 8.8: Commands to be placed into the warpprint environment
Section 9: Special cases and limitations

Text is not converting:

- Font-related UTF-8 information must be embedded in the PDF file. See section 8.2 regarding vector fonts.

Undefined HTML settings:

- See the warning regarding the placement of the HTML settings at section 8.4.

Tabular problems: See section 9.9.

Obscure error messages:

Print first: Be sure that a print version of the document compiles and that your document's LaTeX code is correct, before attempting to generate an HTML version.

\end{warpHTML}, \end{warpprint}, \end{warpall}: Each of these must be without any other characters on the same line.

Options clash: If using **memoir**, see section 9.12.

"No room for a new \write.": Before \usepackage{lwarp}, add:

```
\usepackage{morewrites}
\morewritessetup{allocate=10}
```

"Missing $ inserted.": If using a filename or URL in a footnote or \item, escape underscores with _.

"Label(s) may have changed. Rerun to get cross-references right.":

This warning may repeat endlessly if a math expression is used in a caption. Simple math expressions such as $X=1$ may be replaced with

```
\textit{X}\,=\,1
```

"Leaders not followed by proper glue": This can be caused by a missing l@<floattype> or l@<sectiontype> definition. See **lwarp**'s definitions for examples.

"Improper \prevdepth": lateximages and SVG math require \newpage, which cannot work inside TEX boxes or \ensuremath. Anything using \newsavebox, \newbox, \lrbox, \savebox, \hbox, \vbox, \usebox, \sbox, etc., must be modified to work without box commands.

If you find something using \ensuremath, have it temporarily set:

```
\LetLtxMacro\@ensuredmath\LWR@origensuredmath
```

inside a group first.

⚠ custom macros in section names

Also, custom macros which appear inside a section, figure, or table name should be made robust since they appear inside the .toc, .lof, or .lot files. Use \newrobustcmd or \robustify from **etoolbox**, **xparse**, etc.

⚠ BibTeX

If using BibTeX, see section 9.5.9.

⚠ **polyglossia**

"! Undefined control sequence. ... \@begindocumenthook": See section 9.13.4 if using **polyglossia**.

⚠ custom macros for environments

"\begin{equation} ended by \end{document}": Do not use custom macros such as \beq and \eeq to replace

```
\begin{equation}
...
\end{equation}
```

⚠ \LWR@formatted

"Misplaced \omit": If using \LWR@formatted to define new macros for print and HTML modes, see section 33 regarding \LWR@expandableformatted.

⚠ display math

Complicated objects inside display math: Some objects, such as Ti*k*z, may not compile in **lwarp**'s normal display math emulation. Insert

```
\displaymathother
```

before the display math environment, and then

```
\displaymathnormal
```

when displaying "normal" math. See section 9.6.8.

⚠ MathJax

Incorrect MATHJAX: Some objects do not convert to MATHJAX. Use \displaymathother before these objects, then \displaymathnormal to return to "normal" display math. See section 9.6.8.

Missing sections: See section 8.4 regarding the FileDepth and SideTOCDepth counters, and the use of \tableofcontents in the home page.

Misnumbered footnotes from section headings: See section 9.4.4.

Missing HTML files:

- See the warning regarding changes to the HTML settings at section 8.4.
- Ensure that the filenames are unique after math and short words are removed. See FileSectionNames at section 8.4.

Missing / incorrect cross-references:

- Use `lwarpmk again` followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.
- Labels with special characters may be a problem. It is best to stick with alpha-numeric, hyphen, underscore, and perhaps the colon (if not French).

  `\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.
- **cleveref** and **varioref** are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to "for".

  Ex:

  ```
  \cpageref{tab:first,tab:second}
  ```

  in HTML becomes:

  "pages **for** table 4.1 and **for** table 4.2"

  See `\cpagerefFor` at page 539 to redefine the message which is printed for page number references.

labels

⚠ underscores

`\nameref`

⚠ empty link

⚠ **cleveref** page numbers

BibTeX errors with `\etalchar`: See section 9.5.9.

Malformed URLs: Do not use the % character between arguments of `\hyperref`, etc., as this character is among those which is neutralized for inclusion in HTML URLs.

Em-dashes or En-dashes in listing captions and titles:

Use XeLaTeX or LuaLaTeX.

Floats out of sequence:

Mixed "Here" and floating: Floats [H]ere and regular floats may become out of order. `\clearpage` if necessary.

Caption setup: With `\captionsetup` set the positions for the captions `above` or `below` to match their use in the source code.

Images are appearing in strange places:

- Enter ***`lwarpmk limages`*** to refresh the `lateximage` images.

SVG images:

⚠ adding/removing

When a math expression, `picture`, or Ti*k*z environment is added or removed, the SVG images must be re-created by entering ***`lwarpmk limages`*** to maintain the proper image-file associations. Inline SVG math may be hashed and thus not need to be recreated, but display math and objects such as Ti*k*z may move to new image numbers when the document is changed.

Before attempting to create the SVG image files, **lwarpmk** verifies that the HTML version of the document exists and has correct internal image references.[15] If it is necessary to recompile the document's HTML version, **lwarpmk** will inform so with an error message.

⚠ HTML instead of images

If HTML appears where an SVG image should be, recompile the document one more time to get the page numbers back in sync, then remake the images one more time.

⚠ page counter

Incorrect SVG images will also occur if the document changes the `page` counter:

```
\setcounter{page}{<value>}
```

The `page` counter must *not* be adjusted by the user.

⚠ Lots of files!

Expressing math as SVG images has the advantage of representing the math exactly as LaTeX would, but has the disadvantage of requiring an individual file for each math expression. For inline math, and some other objects, **lwarp** uses an MD5 hash on its LaTeX source to combine multiple instances of identical inline expressions into a single image file, but display math and other environments such as `picture` and Ti*k*z require one image file each. For a document with a large amount of math, see section 6.5 to use MathJax instead.

Plain-looking document:

- The document's CSS stylesheet may not be available, or may be linked incorrectly. Verify any `\CSSFilename` statements point to a valid CSS file.

Broken fragments of HTML:

- Check the PDF file used to create HTML to see if the tags overflowed the margin. (This is why such large page size and margins are used.)

Changes do not seem to be taking effect:

- Be sure to `lwarpmk clean`, recompile, then start by reloading the home page. You may have been looking at an older version of the document. If you changed a section name, you may have been looking at the file for the old name.
- See the warning regarding changes to the HTML settings at section 8.4.
- Verify that the proper CSS is actually being used.
- The browser may compensate for some subtle changes, such as automatically generating ligatures, reflowing text, etc.

Un-matched conditional compiles:

- Verify the proper `begin/end` of `warpprint`, `warpHTML`, and `warpall` environments.

[15] This becomes important when dealing with a document containing thousands of images.

13.1.1 Debug tracing output

\tracinglwarp When \tracinglwarp is used, **lwarp** will add extra tracing messages to the .log file. The last several messages may help track down errors.

Place \tracinglwarp just after \usepackage{lwarp} to activate tracing.

13.2 Compiling the lwarp.dtx file

lwarp_tutorial.tex: Copy or link lwarp_tutorial.txt from the TDS doc directory to the source directory, or wherever you wish to compile the documentation. This file is included verbatim in the documentation, but is in the doc directory so that it may be found by **texdoc** and copied by the user.

Illogical error messages caused by an out-of-sync lwarp.sty file:

1. Delete the lwarp.sty file.
2. Enter ***pdflatex lwarp.ins*** to generate a new lwarp.sty file.
3. Enter ***pdflatex lwarp.dtx*** to recompile the lwarp.pdf documentation.

Un-nested environments:

Be sure to properly nest:

- \begin{macrocode} and \end{macrocode}
- \begin{macro} and \end{macro}
- \begin{environment} and \end{environment}

File 1 `lwarp.sty`

14 Implementation

This package is perhaps best described as a large collection of smaller individual technical challenges, in many cases solved through a number of ~~crude hacks~~ clever tricks. Reference sources are given for many of the solutions, and a quick internet search will provide additional possibilities.

Judgement calls were made, and are often commented. Improvements are possible. The author is open to ideas and suggestions.

Packages were patched for re-use where they provided significant functionality. Examples include **xcolor** with its color models and conversion to HTML color output, and **siunitx** which provides many number and unit-formatting options, almost all of which are available in pure-text form, and thus easily used by **pdftotext**.

Packages were emulated where their primary purpose was visual formatting which is not relevent to HTML output. For example, packages related to sectioning are already patched by numerous other packages, creating a difficult number of combinations to try to support, and yet in HTML output all of the formatting is thrown away, so these packages are merely emulated.

Packages with graphical output are allowed as-is, but must be nested inside a `lateximage` environment to preserve the graphics.

Testing has primarily been done with the Iceweasel/Firefox browser.

Table 7: Section depths and HTML headings

Section	LaTeX depth	HTML headings *
title of the entire website		`<h1>`
none	-5	new for this package
book	-2	**not yet used**
part	-1	`<h2>`
chapter	0	`<h3>`
section	1	`<h4>`
subsection	2	`<h5>`
subsubsection	3	`<h6>`
paragraph	4	`<span class = "paragraph">`
subparagraph	5	`<span class = "subparagraph">`
listitem	7	new for this package, used for list items

* If `FormatWP` is true, section headings may be adjusted, depending on `WPTitleHeading`. See table 6 on page 169.

15 Section depths and HTML headings

Stacks are created to track depth inside the LaTeX document structure. This depth is translated to HTML headings as shown in table 7. "Depth" here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the LaTeX document structure.

When starting a new section, the program first must close out any existing sections and lists of a deeper level to keep the HTML tags nested correctly.

Support for the **memoir** package will require the addition of a `book` level, which may push the HTML headings down a step, and also cause `subsubsection` to become a `<div>` due to a limit of six HTML headings.

It is possible to use HTML5 `<section>` and `<h1>` for all levels, but this may not be well-recognized by older browsers.

Fixed levels for parts and chapters allow the CSS to remain fixed as well.

16 Source Code

This is where the documented source code for **lwarp** begins, continuing through the following sections all the way to the change log and index at the end of this document.

The following sections document the actual implementation of the **lwarp** package.

line numbers The small numbers at the left end of a line refer to line numbers in the `lwarp.sty` file.

subjects Blue-colored tags in the left margin aid in quickly identifying the subject of each paragraph.

objects Black-colored tags in the left marign are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are command macros. Each of these also appears in the index as individual entries, and are also listed together under "files", "packages", "environments", "booleans", and "counters".

index entries

⚠ warnings Special warnings are marked with a warning icon.

for HTML output:
for PRINT output:
for HTML & PRINT:

Green-colored tags in the left margin show which sections of source code apply to the generation of HTML, print, or both forms of output.

— **lwarp** source code begins on the following page —

17 Detecting the TEX Engine — pdflatex, lualatex, xelatex

See: http://tex.stackexchange.com/a/47579.

Detects XETEX and LuaLATEX:

```
\RequirePackage{iftex}
\newif\ifxetexorluatex
\ifXeTeX
    \xetexorluatextrue
\else
    \ifLuaTeX
        \xetexorluatextrue
    \else
        \xetexorluatexfalse
    \fi
\fi

\ifLuaTeX
\RequirePackage{luatex85}% until the geometry package is updated
\fi

\RequirePackage{ifpdf}
```

18 MD5 hashing

The MD5 hash is used for `lateximage` filenames for SVG math.

```
\newcommand{\LWR@mdfive}[1]{%
\PackageError{lwarp}
{No MD5 macro was found.}
{Lwarp must find the macros pdfmdfivesum or mdfivesum.}
}

\ifPDFTeX
\let\LWR@mdfive\pdfmdfivesum
\fi

\ifLuaTeX
\RequirePackage{pdftexcmds}
\let\LWR@mdfive\pdf@mdfivesum
\fi

\ifXeTeX
```

```
\@ifundefined{pdffivesum}{}
    {\let\LWR@mdfive\pdfmdfivesum}
\@ifundefined{mdfivesum}{}
    {\let\LWR@mdfive\mdfivesum}
\fi
```

19 pdfLaTeX T1 and UTF8 encoding

When using pdfLaTeX, **lwarp** requires T1 encoding, and recommends UTF8 encoding.

If some other input encoding is already defined, **lwarp** will try to use it instead, and hope for the best.

XeLaTeX and LuaLaTeX are both UTF8 by nature.

```
\ifPDFTeX
\RequirePackage[T1]{fontenc}

\@ifpackageloaded{inputenc}{}{
    \@ifpackageloaded{inputenx}{}{
        \RequirePackage[utf8]{inputenc}
    }
}
\fi
```

20 Unicode input characters

for HTML & PRINT:

If using **pdflatex**, convert a minimal set of Unicode characters. Additional characters may be defined by the user, as needed.

A commonly-used multiply symbol is declared to be `\texttimes`.

The first arguments of `\newunicodechar` below are text ligatures in the source code, even though they are not printed in the following listing.

```
\RequirePackage{newunicodechar}

\newunicodechar{×}{\texttimes}

\ifPDFTeX
\newunicodechar{ff}{ff}% the first arguments are ligatures
\newunicodechar{fi}{fi}
\newunicodechar{fl}{fl}
```

```
\newunicodechar{ffi}{ffi}
\newunicodechar{ffl}{ffl}
\newunicodechar{--}{---}
\newunicodechar{-}{--}
```

In PDFTEX, preserve upright quotes in verbatim text:

```
\RequirePackage{upquote}
\else
\fi
```

21 Miscellaneous tools

`\LWR@providelength` {⟨`\lengthname`⟩} Provides the length if it isn't defined yet.

Used to provide source compatibility for lengths which will be ignored, but might or might not be already provided by other packages.

```
\newcommand*\LWR@providelength[1]{%
    \ifdeflength{#1}{}{\newlength{#1}}%
}
```

Prints a length in the given units, without printing the unit itself.

`\LWR@convertto` {⟨*dest unit*⟩} {⟨*length*⟩}

```
\newcommand*{\LWR@convertto}[2]{\strip@pt\dimexpr #2*65536/\number\dimexpr 1#1}
```

`\LWR@patcherror` {⟨*packagename*⟩} {⟨*macroname*⟩}

Prints an error if could not patch a macro.

```
\newcommand*{\LWR@patcherror}[2]{%
\PackageError{lwarp}
{Unable to patch package #1, macro #2}
{Please contact the author of the lwarp package.}
}
```

22 Early package requirements

Pkg `etoolbox` Provides `\ifbool` and other functions.

Pkg xpatch Patches macros with optional arguments.

```
\RequirePackage{etoolbox}[2011/01/03]% v2.6 for \BeforeBeginEnvironment, etc.
\RequirePackage{xpatch}
```

Pkg ifplatform Provides \ifwindows to try to automatically detect WINDOWS OS.

```
\RequirePackage{ifplatform}% sense op-system platform
```

Pkg letltxmacro Used to redefine \textbf and friends.

```
\RequirePackage{letltxmacro}
```

23 Operating-System portability

Prog Unix
Prog Mac OS
Prog Linux
Prog MS-Windows
Prog Windows
Opt OSWindows

lwarp tries to detect which operating system is being used. UNIX / MAC OS / LINUX is the default (collectively referred to as "UNIX" in the configuration files), and MS-WINDOWS is supported as well.

If MS-WINDOWS is not correctly detected, use the **lwarp** option OSWindows.

When detected or specified, the operating-system path separator used by **lwarp** is modified, and the boolean usingOSWindows is set true. This boolean may be tested by the user for later use.

23.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used, or if WINDOWS is automatically detected.

```
\newbool{usingOSWindows}
\boolfalse{usingOSWindows}
```

23.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

```
\newcommand*{\OSPathSymbol}{/}
```

23.3 MS-Windows

For MS-Windows:

\LWR@setOSWindows Set defaults for the MS-Windows operating system. **lwarp** attempts to auto-detect the operatings system, and the `OSWindows` option may also be used to force MS-Windows compatibility.

```
\newcommand*{\LWR@setOSWindows}
{
\booltrue{usingOSWindows}
\renewcommand*{\OSPathSymbol}{\@backslashchar}
}
```

Test for windows during compile. The user may also specify `OSWindows` package option in case this test fails.

```
\ifwindows
\LWR@setOSWindows
\fi
```

24 Package options

Pkg kvoptions Allows key/value package options.

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{family=LWR,prefix=LWR@}
```

Bool warpingprint
Bool warpingHTML
Bool mathjax
Bool LWR@origmathjax

Set to true/false depending on the package option selections for print/HTML/EPUB output and mathsvg/mathjax.

`LWR@origmathjax` remembers the original setting to be restored by `\displaymathnormal`.

```
\newbool{warpingprint}
\newbool{warpingHTML}
\newbool{mathjax}
\newbool{LWR@origmathjax}
```

defaults The default is print output, and SVG math if the user chose HTML output.

```
\booltrue{warpingprint}%
```

```
\boolfalse{warpingHTML}%
\boolfalse{mathjax}%
```

Opt warpprint If the warpprint option is given, boolean warpingprint is true and boolean warpingHTML is false, and may be used for \ifbool tests.

```
\DeclareVoidOption{warpprint}{%
\PackageInfo{lwarp}{Using option 'warpprint'}
\booltrue{warpingprint}%
\boolfalse{warpingHTML}%
}
```

Env warpHTML Anything in the warpHTML environment will be generated for HTML output only.

Opt warpHTML If the warpHTML option is given, boolean warpingHTML is true and boolean warpingprint is false, and may be used for \ifbool tests.

```
\DeclareVoidOption{warpHTML}{%
\PackageInfo{lwarp}{Using option 'warpHTML'}%
\booltrue{warpingHTML}%
\boolfalse{warpingprint}%
}
```

Opt mathsvg Option mathsvg selects SVG math display: If the mathsvg option is given, boolean mathjax is false, and may be used for \ifbool tests.

```
\DeclareVoidOption{mathsvg}{%
\PackageInfo{lwarp}{Using option 'mathsvg'}
\boolfalse{mathjax}%
\boolfalse{LWR@origmathjax}%
}
```

Opt mathjax Option mathjax selects MathJax math display: If the mathjax option is given, boolean mathjax is true, may be used for \ifbool tests.

```
\DeclareVoidOption{mathjax}{%
\PackageInfo{lwarp}{Using option 'mathjax'}
\booltrue{mathjax}%
\booltrue{LWR@origmathjax}%
}
```

Opt BaseJobname Option BaseJobname sets the \BaseJobname for this document.

This is the \jobname of the printed version, even if currently compiling the HTML version. I.e. this is the \jobname without _html appended. This is used to set \HomeHTMLFilename if the user did not provide one.

```
\DeclareStringOption[\jobname]{BaseJobname}
```

Opt `makeindexStyle` Selects a custom `.ist` file. The default is `lwarp.ist`. A customized file should be based on `lwarp.ist`, and must retain the lines related to `\hyperindexref`.

```
\DeclareStringOption[lwarp.ist]{makeindexStyle}
```

Opt `xindyStyle` Selects a custom `.xdy` file. The default is `lwarp.xdy`. A customized file should be based on `lwarp.xdy`, and must retain the line

```
(markup-locref :open "\hyperindexref{" :close "}")
```

```
\DeclareStringOption[lwarp.xdy]{xindyStyle}
```

Opt `xindyLanguage` Sets the **xindy** language to be assigned in **lwarpmk**'s configuration files. This is then used by **lwarpmk** while processing the index and glossary.

```
\DeclareStringOption[english]{xindyLanguage}
```

Opt `xindyCodepage` Sets the **xindy** codepage to be assigned in **lwarpmk**'s configuration files. This is then used by **lwarpmk** while processing the index.

```
\DeclareStringOption[utf8]{xindyCodepage}
```

Opt `pdftotextEnc` The option `pdftotextEnc` sets the encoding used by **pdftotext**. This is passed to **pdftotext** using its `-enc` option, and is used when converting LaTeX PDF output with HTML tags into a plain-text file with HTML tags.
Default: `UTF-8`

```
\DeclareStringOption[UTF-8]{pdftotextEnc}
```

Opt `lwarpmk` Tells **lwarp** to generate a local copy of **lwarpmk** called `lwarpmk.lua`. Useful for archiving for future use. This file may be made executable and acts just like **lwarpmk**.

If `lwarpmk` option, creates a local copy of `lwarpmk.lua`:

```
\newbool{LWR@creatinglwarpmk}
\boolfalse{LWR@creatinglwarpmk}

\DeclareVoidOption{lwarpmk}{
\PackageInfo{lwarp}{Using option 'lwarpmk'}
\booltrue{LWR@creatinglwarpmk}
}
```

Opt `OSWindows` Tells **lwarp** to use MS-Windows compatibility. Auto-detection of the operating sys-

tem is attempted, and this option is only necessary if the auto-detection fails. See the automatically-generated `lwarpmk.conf` file to find out whether the operating system was detected correctly.

```
\DeclareVoidOption{OSWindows}{
\PackageInfo{lwarp}{Using option 'OSWindows'}
\LWR@setOSWindows
}
```

Opt `HomeHTMLFilename` Default: `\lwarp`

The filename of the homepage. The default is the jobname. This option is stored into `\LWR@HomeHTMLFilename`, and later transferred into `\HomeHTMLFilename` for internal use.

```
\DeclareStringOption[]{HomeHTMLFilename}
```

Opt `HTMLFilename` Default: `<empty>`

The filename prefix of web pages after the homepage. The default is empty, no prefix. This option is stored into `\LWR@HTMLFilename`, and later transferred into `\HTMLFilename` for internal use.

```
\DeclareStringOption[]{HTMLFilename}
```

Opt `PrintIndexCmd` Default: `<empty>`

The shell commands to use to compile the print indexes.

```
\DeclareStringOption[]{PrintIndexCmd}
```

Opt `HTMLIndexCmd` Default: `<empty>`

The shell commands to use to compile the HTML indexes.

```
\DeclareStringOption[]{HTMLIndexCmd}
```

Opt `LatexmkIndexCmd` Default: `<empty>`

The shell commands to by used by **latexmk** to compile the print indexes. Unlike `PrintIndexCmd` and `HTMLIndexCmd`, `LatexmkIndexCmd` does not include the filename, which will be provided by **latexmk**.

```
\DeclareStringOption[]{LatexmkIndexCmd}
```

Opt `makeindex`

Tells **lwarp** to use **makeindex** for index generation. When `lwarpmk.conf` and `*.lwarpmkconf` are generated, `PrintIndexCmd` and `HTMLIndexCmd` will be set for **makeindex** with a single index file.

```
\DeclareBoolOption[false]{makeindex}
```

Opt `xindy`

Tells **lwarp** to use **xindy** for index generation. When `lwarpmk.conf` and `*.lwarpmkconf` are generated, `PrintIndexCmd` and `HTMLIndexCmd` will be set for **xindy** with a single index file.

```
\DeclareBoolOption[false]{xindy}
```

Opt GlossaryCmd
Default: makeglossaries

The shell command to use to compile the glossary. The print or HTML version of the glossary filename will be appended to this command.

```
\DeclareStringOption[makeglossaries]{GlossaryCmd}
```

Opt latexmk

Option latexmk tells **lwarpmk** to use **latexmk** when compiling documents.

```
\DeclareBoolOption[false]{latexmk}
```

Opt dvips

Option dvips tells **lwarpmk** to use **dvips** when compiling DVI **latex** documents.

```
\DeclareBoolOption[false]{dvips}
```

Opt dvipdfm

Option dvipdfm tells **lwarpmk** to use **dvipdfm** when compiling DVI **latex** documents.

```
\DeclareBoolOption[false]{dvipdfm}
```

Opt dvipdfmx

Option dvipdfmx tells **lwarpmk** to use **dvipdfmx** when compiling DVI **latex** documents.

```
\DeclareBoolOption[false]{dvipdfmx}
```

Execute options

Execute the package options, with the defaults which have been set just above:

```
\ProcessKeyvalOptions*\relax
```

Assign the \BaseJobname if the user hasn't provided one:

```
\providecommand*{\BaseJobname}{\LWR@BaseJobname}
```

Defaults unless already over-ridden by the user:

```
\ifcsempty{LWR@HomeHTMLFilename}{
\newcommand*{\HomeHTMLFilename}{\BaseJobname}
}{
\csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
}

\csedef{HTMLFilename}{\LWR@HTMLFilename}
```

\LWR@PrintIndexCmd and \LWR@HTMLIndexCmd are tested to see if they are empty. If so, they are set to a reasonable defaults for a single index using **makeindex**, then possibly set to defaults for **xindy** if the **lwarp** xindy option was selected.

```
\ifdefempty{\LWR@PrintIndexCmd}{
```

```
    \renewcommand{\LWR@PrintIndexCmd}{%
        makeindex  -s \LWR@makeindexStyle \space  \jobname.idx%
    }
    \ifbool{LWR@xindy}{
        \renewcommand{\LWR@PrintIndexCmd}{%
            xindy
            -M \LWR@xindyStyle \space
            -L \LWR@xindyLanguage \space
            -C \LWR@xindyCodepage \space
            \jobname.idx%
        }
    }{}
}{}

\ifdefempty{\LWR@HTMLIndexCmd}{
    \renewcommand{\LWR@HTMLIndexCmd}{%
        makeindex  -s \LWR@makeindexStyle \space  \jobname_html.idx%
    }
    \ifbool{LWR@xindy}{
        \renewcommand{\LWR@HTMLIndexCmd}{%
            xindy
            -M \LWR@xindyStyle \space
            -L \LWR@xindyLanguage \space
            -C \LWR@xindyCodepage \space
            \jobname_html.idx%
        }
    }{}
}{}

\ifdefempty{\LWR@LatexmkIndexCmd}{
    \renewcommand{\LWR@LatexmkIndexCmd}{%
        makeindex  -s \LWR@makeindexStyle%
    }
    \ifbool{LWR@xindy}{
        \renewcommand{\LWR@LatexmkIndexCmd}{%
            xindy
            -M \LWR@xindyStyle \space
            -L \LWR@xindyLanguage \space
            -C \LWR@xindyCodepage%
        }
    }{}
}{}
```

24.1 Conditional compilation

`\warpprintonly` {⟨*contents*⟩}

Only process the contents if producing printed output.

```
\newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}
```

`\warpHTMLonly` {⟨*contents*⟩}

Only process the contents if producing HTML output.

```
\newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}
```

Pkg `comment` Provides conditional code blocks.

Attempts to use **versions** or **verbatim** fail in some cases, and do not provide much of a speed benefit even when they do work.

```
\RequirePackage{comment}
```

Use `comment_print.cut` for print mode, and `comment_html.cut` for HTML mode. This helps **latexmk** to more reliably know whether to recompile.

```
\ifbool{warpingHTML}{
\def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
}{}

\ifbool{warpingprint}{
\def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
}{}

\excludecomment{testing}
```

Env `warpall` Anything in the `warpall` environment will be generated for print or HTML outputs.

```
\includecomment{warpall}
```

Env `warpprint` Anything in the `warpprint` environment will be generated for print output only.

Env `warpHTML`

For HTML output:

```
\ifbool{warpingHTML}{%
\includecomment{warpHTML}
}
{\excludecomment{warpHTML}}%

\ifbool{warpingprint}
{\includecomment{warpprint}}
{\excludecomment{warpprint}}
```

Optionally generate a local copy of **lwarpmk**. Default to no.

```
\ifbool{LWR@creatinglwarpmk}
{\includecomment{LWR@createlwarpmk}}
{\excludecomment{LWR@createlwarpmk}}
```

25 Package load order

Several packages should only be loaded before **lwarp**, and most others should only be loaded after.

Packages which should only be loaded before **lwarp** have their own

```
lwarp-<packagename>.sty
```

which use `\LWR@loadbefore` to trigger an error if they are loaded after **lwarp**. Examples include **fontspec**, **inputenc**, **inputenx**, **fontenc**, and **newunicodechar**.

Most packages should be loaded after **lwarp**. This is enfoced by a large number of `\LWR@loadafter` statements, below.

Some packages are emulated by **memoir**, and so these are tested by `\LWR@notmemoirloadafter`, which does not cause an error if **memoir** is used.

25.1 Tests of package load order

`\LWR@loadafter` {⟨*packagename*⟩} Error if this package was loaded before **lwarp**.

```
\newcommand*{\LWR@loadafter}[1]{%
\@ifpackageloaded{#1}
{
\PackageError{lwarp}
{Package #1, or one which uses #1, must be loaded after lwarp}
{Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
Package #1 may also be loaded by something else, which must also be moved
after lwarp.}
}
{}
}
```

`\LWR@notmemoirloadafter` {⟨*packagename*⟩} Error if not **memoir** class and this package was loaded before **lwarp**.

memoir emulates many packages, and pretends that they have already been loaded.

```
\@ifclassloaded{memoir}
{\newcommand*{\LWR@notmemoirloadafter}[1]{}}
{\LetLtxMacro\LWR@notmemoirloadafter\LWR@loadafter}
```

`\LWR@loadbefore` {⟨*packagename*⟩} Error if this package is after **lwarp**.

```
\newcommand*{\LWR@loadbefore}[1]{%
\@ifpackageloaded{#1}
{}
{
\PackageError{lwarp}
{Package #1 must be loaded before lwarp}
{Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
}
}
```

`\LWR@loadnever` {⟨*badpackagename*⟩} {⟨*replacementpkgname*⟩}

The first packages is not supported, so tell the user to use the second instead.

```
\newcommand*{\LWR@loadnever}[2]{%
\PackageError{lwarp}
{Package #1 is not supported by lwarp's HTML conversion.
Package(s) #2 may be useful instead}
{Package #1 might conflict with lwarp in some way,
or is superceded by another package.
For a possible alternative, see package(s) #2.}
}
```

`\LWR@earlyloadnever` {⟨*badpackagename*⟩} {⟨*replacementpkgname*⟩}

The first packages is not supported, so tell the user to use the second instead. This version checks immediately for packages which may have been loaded before **lwarp**.

```
\newcommand*{\LWR@earlyloadnever}[2]{%
\@ifpackageloaded{#1}{%
\PackageError{lwarp}
{Package #1 is not supported by lwarp's HTML conversion.
Package(s) #2 may be useful instead}
{Package #1 might conflict with lwarp in some way,
or is superceded by another package.
For a possible alternative, see package(s) #2.}
}{}%
}
```

25.2 Error for disallowed packages loaded before lwarp

```
\LWR@earlyloadnever{ae}{lmodern}
\LWR@earlyloadnever{aecc}{lmodern}
\LWR@earlyloadnever{boxedminipage}{boxedminipage2e}
\LWR@earlyloadnever{caption2}{caption}
% \LWR@earlyloadnever{ccaption}{caption}% might be preloaded by memoir
\LWR@earlyloadnever{fancyheadings}{fancyhdr}
\LWR@earlyloadnever{glossary}{glossaries}
\LWR@earlyloadnever{t1enc}{fontenc, inputenc, inputenx}
\LWR@earlyloadnever{wasysym}{textcomp, amssymb, amsfonts, mnsymbol, fdsymbol}
```

25.3 Enforcing package loading after lwarp

Packages which should only be loaded after **lwarp** are tested here to trip an error of they have already been loaded.

The following packages must be loaded after **lwarp**:

```
\LWR@loadafter{2up}
\LWR@loadafter{a4}
\LWR@loadafter{a4wide}
\LWR@loadafter{a5comb}
\LWR@notmemoirloadafter{abstract}
\LWR@loadafter{accsupp}
\LWR@loadafter{acro}
\LWR@loadafter{acronym}
\LWR@loadafter{adjmulticol}
\LWR@loadafter{addlines}
\LWR@loadafter{ae}
\LWR@loadafter{aecc}
\LWR@loadafter{afterpage}
\LWR@loadafter{algorithm2e}
\LWR@loadafter{algorithmicx}
\LWR@loadafter{alltt}
\LWR@loadafter{amsmath}
\LWR@loadafter{amsthm}
\LWR@loadafter{anonchap}
\LWR@loadafter{anysize}
\LWR@notmemoirloadafter{appendix}
\LWR@loadafter{arabicfront}
\LWR@notmemoirloadafter{array}
\LWR@loadafter{arydshln}
\LWR@loadafter{asymptote}
% \LWR@loadafter{atbegshi}% used by morewrites
\LWR@loadafter{attachfile}
\LWR@loadafter{attachfile2}
\LWR@loadafter{authblk}
```

```
\LWR@loadafter{axessibility}
\LWR@loadafter{axodraw2}
\LWR@loadafter{backref}
\LWR@loadafter{balance}
\LWR@loadafter{bigdelim}
\LWR@loadafter{bigstrut}
\LWR@loadafter{blowup}
\LWR@loadafter{booklet}
\LWR@loadafter{bookmark}
\LWR@notmemoirloadafter{booktabs}
\LWR@loadafter{bophook}
\LWR@loadafter{boxedminipage}
\LWR@loadafter{boxedminipage2e}
\LWR@loadafter{breakurl}
\LWR@loadafter{breqn}
\LWR@loadafter{bxpapersize}
\LWR@loadafter{bytefield}
\LWR@loadafter{cancel}
\LWR@loadafter{canoniclayout}
\LWR@loadafter{caption}
\LWR@loadafter{caption2}
\LWR@loadafter{cases}
% \LWR@loadafter{ccaption}% may be preloaded by memoir
\LWR@loadafter{changebar}
\LWR@notmemoirloadafter{changepage}
\LWR@notmemoirloadafter{chngpage}
\LWR@loadafter{chappg}
\LWR@loadafter{chapterbib}
\LWR@loadafter{chemfig}
\LWR@loadafter{chemformula}
\LWR@loadafter{chemgreek}
\LWR@loadafter{chemmacros}
\LWR@loadafter{chemnum}
\LWR@loadafter{cite}
\LWR@loadafter{color}
\LWR@loadafter{colortbl}
\LWR@loadafter{continue}
\LWR@notmemoirloadafter{crop}
\LWR@loadafter{cuted}
\LWR@loadafter{cutwin}
\LWR@loadafter{dblfloatfix}
\LWR@loadafter{dblfnote}
\LWR@notmemoirloadafter{dcolumn}
\LWR@loadafter{diagbox}
\LWR@loadafter{draftcopy}
\LWR@loadafter{draftfigure}
\LWR@loadafter{draftwatermark}
\LWR@loadafter{easy-todo}
\LWR@loadafter{ebook}
\LWR@loadafter{ellipsis}
```

```
\LWR@loadafter{emptypage}
\LWR@loadafter{endfloat}
\LWR@loadafter{endheads}
\LWR@loadafter{endnotes}
\LWR@notmemoirloadafter{enumerate}
\LWR@loadafter{enumitem}
\LWR@notmemoirloadafter{epigraph}
\LWR@loadafter{epstopdf}
\LWR@loadafter{epstopdf-base}
\LWR@loadafter{errata}
\LWR@loadafter{eso-pic}
\LWR@loadafter{everypage}
\LWR@loadafter{everyshi}
\LWR@loadafter{extramarks}
\LWR@loadafter{fancybox}
\LWR@loadafter{fancyhdr}
\LWR@loadafter{fancyheadings}
\LWR@loadafter{fancyref}
\LWR@loadafter{fancytabs}
\LWR@loadafter{fancyvrb}
\LWR@loadafter{figcaps}
\LWR@loadafter{figsize}
\LWR@loadafter{fix2col}
\LWR@loadafter{fixme}
\LWR@loadafter{fixmetodonotes}
\LWR@loadafter{flafter}
\LWR@loadafter{float}
\LWR@loadafter{floatflt}
\LWR@loadafter{floatpag}
\LWR@loadafter{floatrow}
\LWR@loadafter{fltrace}
\LWR@loadafter{flushend}
\LWR@loadafter{fnbreak}
\LWR@loadafter{fncychap}
\LWR@loadafter{fnlineno}
\LWR@loadafter{fnpos}
% fontenc must be loaded before lwarp
% fontspec must be loaded before lwarp
\LWR@loadafter{footmisc}
\LWR@loadafter{footnote}
\LWR@loadafter{footnotehyper}
\LWR@loadafter{footnpag}
\LWR@loadafter{forest}
\LWR@loadafter{framed}
\LWR@loadafter{ftnright}
\LWR@loadafter{fullminipage}
\LWR@loadafter{fullpage}
\LWR@loadafter{fullwidth}
\LWR@loadafter{fwlw}
\LWR@loadafter{geometry}
```

```
\LWR@loadafter{glossaries}
% \LWR@loadafter{graphics}% pre-loaded by xunicode
% \LWR@loadafter{graphicx}% pre-loaded by xunicode
\LWR@loadafter{glossary}
\LWR@loadafter{grffile}
\LWR@loadafter{grid}
\LWR@loadafter{grid-system}
\LWR@loadafter{gridset}
\LWR@loadafter{hang}
\LWR@loadafter{hanging}
\LWR@loadafter{hypcap}
\LWR@loadafter{hypdestopt}
\LWR@loadafter{hypernat}
\LWR@loadafter{hyperref}
\LWR@loadafter{hyperxmp}
\LWR@loadafter{hyphenat}
\LWR@loadafter{idxlayout}
\LWR@loadafter{ifoddpage}
\LWR@loadafter{imakeidx}
\LWR@loadafter{indentfirst}
\LWR@notmemoirloadafter{index}
% inputenc must be loaded before lwarp
% inputenx must be loaded before lwarp
\LWR@loadafter{intopdf}
\LWR@loadafter{keyfloat}
\LWR@loadafter{layaureo}
\LWR@loadafter{layout}
\LWR@loadafter{leading}
\LWR@loadafter{letterspace}
\LWR@loadafter{lettrine}
\LWR@loadafter{lineno}
\LWR@loadafter{lips}
\LWR@loadafter{listings}
\LWR@loadafter{longtable}
\LWR@loadafter{lscape}
\LWR@loadafter{ltablex}
\LWR@loadafter{ltcaption}
\LWR@loadafter{ltxgrid}
\LWR@loadafter{ltxtable}
\LWR@loadafter{lua-check-hyphen}
\LWR@loadafter{luacolor}
\LWR@loadafter{luatodonotes}
\LWR@loadafter{magaz}
\LWR@notmemoirloadafter{makeidx}
\LWR@loadafter{marginfit}
\LWR@loadafter{marginfix}
\LWR@loadafter{marginnote}
\LWR@loadafter{mcaption}
\LWR@loadafter{mdframed}
\LWR@loadafter{memhfixc}
```

```
\LWR@loadafter{metalogo}
\LWR@loadafter{mhchem}
\LWR@loadafter{microtype}
\LWR@loadafter{midfloat}
\LWR@loadafter{midpage}
\LWR@loadafter{morefloats}
\LWR@notmemoirloadafter{moreverb}
% morewrites must be loaded before lwarp
\LWR@notmemoirloadafter{mparhack}
%\LWR@loadafter{multicol}% loaded by ltxdoc
\LWR@loadafter{multirow}
\LWR@loadafter{multitoc}
\LWR@loadafter{nameref}
\LWR@loadafter{natbib}
\LWR@notmemoirloadafter{nccfancyhdr}
\LWR@notmemoirloadafter{needspace}
% newclude must be loaded before lwarp
\LWR@loadafter{newtxmath}
% newunicodechar must be loaded before lwarp
\LWR@notmemoirloadafter{nextpage}
\LWR@loadafter{nicefrac}
\LWR@loadafter{nonfloat}
\LWR@loadafter{nonumonpart}
\LWR@loadafter{nopageno}
\LWR@loadafter{nowidow}
\LWR@loadafter{ntheorem}
\LWR@loadafter{overpic}
\LWR@loadafter{pagegrid}
\LWR@notmemoirloadafter{pagenote}
\LWR@loadafter{pagesel}
\LWR@loadafter{paralist}
\LWR@loadafter{parnotes}
\LWR@notmemoirloadafter{parskip}
\LWR@loadafter{pbox}
\LWR@loadafter{pdfrender}
\LWR@loadafter{pdflscape}
\LWR@loadafter{pdfpages}
\LWR@loadafter{pdfsync}
\LWR@loadafter{pdftricks}
\LWR@loadafter{pdfx}
\LWR@loadafter{pfnote}
\LWR@loadafter{phfqit}
\LWR@loadafter{placeins}
\LWR@loadafter{prelim2e}
\LWR@loadafter{prettyref}
\LWR@loadafter{preview}
\LWR@loadafter{pst-eps}
\LWR@loadafter{pstricks}
\LWR@loadafter{quotchap}
\LWR@loadafter{quoting}
```

```
\LWR@loadafter{ragged2e}
\LWR@loadafter{realscripts}
\LWR@loadafter{register}
\LWR@loadafter{relsize}
\LWR@loadafter{repeatindex}
\LWR@loadafter{resizegather}
\LWR@loadafter{romanbar}
\LWR@loadafter{romanbarpagenumber}
\LWR@loadafter{rotating}
\LWR@loadafter{rotfloat}
\LWR@loadafter{savetrees}
% \LWR@loadafter{scalefnt}% loaded by babel-french
\LWR@loadafter{schemata}
\LWR@loadafter{scrextend}
\LWR@loadafter{scrhack}
\LWR@loadafter{scrlayer}
\LWR@loadafter{scrlayer-notecolumn}
\LWR@loadafter{scrlayer-scrpage}
\LWR@loadafter{section}
\LWR@loadafter{sectionbreak}
\LWR@loadafter{sectsty}
\LWR@notmemoirloadafter{setspace}
\LWR@loadafter{shadow}
\LWR@notmemoirloadafter{showidx}
\LWR@loadafter{showkeys}
\LWR@loadafter{sidecap}
\LWR@loadafter{sidenotes}
\LWR@loadafter{SIunits}
\LWR@loadafter{siunitx}
\LWR@loadafter{soul}
\LWR@loadafter{soulpos}
\LWR@loadafter{soulutf8}
\LWR@loadafter{splitidx}
\LWR@loadafter{stabular}
\LWR@loadafter{stfloats}
\LWR@loadafter{subfig}
\LWR@loadafter{subfigure}
\LWR@loadafter{supertabular}
\LWR@loadafter{t1inc}
\LWR@loadafter{tabls}
\LWR@notmemoirloadafter{tabularx}
\LWR@loadafter{tabulary}
\LWR@loadafter{textarea}
% \LWR@loadafter{textcomp}% maybe before lwarp with font packages
\LWR@loadafter{textfit}
\LWR@loadafter{textpos}
\LWR@loadafter{theorem}
\LWR@loadafter{threeparttable}
\LWR@loadafter{thumb}
\LWR@loadafter{thumbs}
```

```
\LWR@loadafter{tikz}
\LWR@loadafter{titleps}
\LWR@loadafter{titlesec}
\LWR@loadafter{titletoc}
\LWR@notmemoirloadafter{titling}
% \LWR@loadafter{tocbasic}% preloaded by koma-script classes
\LWR@notmemoirloadafter{tocbibind}
\LWR@loadafter{tocenter}
\LWR@notmemoirloadafter{tocloft}
\LWR@loadafter{tocstyle}
\LWR@loadafter{todo}
\LWR@loadafter{todonotes}
\LWR@loadafter{transparent}
\LWR@loadafter{trimclip}
\LWR@loadafter{trivfloat}
\LWR@loadafter{turnthepage}
\LWR@loadafter{twoup-gen}


% \LWR@loadafter{typearea}% preloaded by koma-script classes
\LWR@loadafter{ulem}
\LWR@loadafter{underscore}
\LWR@loadafter{units}
\LWR@loadafter{upref}
\LWR@loadafter{url}
\LWR@loadafter{varioref}% no lwarp package provided
\LWR@notmemoirloadafter{verse}
\LWR@loadafter{vertbars}
\LWR@loadafter{vmargin}
\LWR@loadafter{vowel}
\LWR@loadafter{vwcol}
\LWR@loadafter{wallpaper}
\LWR@loadafter{wasysym}
\LWR@loadafter{watermark}
\LWR@loadafter{widows-and-orphans}
\LWR@loadafter{wrapfig}
\LWR@loadafter{xcolor}
\LWR@loadafter{xellipsis}
\LWR@loadafter{xfrac}
\LWR@loadafter{xltabular}
\LWR@loadafter{xltxtra}
\LWR@loadafter{xmpincl}
\LWR@loadafter{xpiano}
\LWR@loadafter{xtab}
% xunicode must be loaded before lwarp
\LWR@loadafter{xurl}
\LWR@loadafter{xy}
\LWR@loadafter{zwpagelayout}
```

26 Required packages

These packages are automatically loaded by **lwarp** when generating HTML output. Some of them are also automatically loaded when generating print output, but some are not.

for HTML output:

```
\begin{warpHTML}
```

Load **fontspec** if necessary:

```
\ifxetexorluatex
\@ifpackageloaded{fontspec}{}{
\usepackage[no-math]{fontspec}
}
```

The monospaced font is used for HTML tags, so turn off its TeX ligatures and common ligatures:

```
\defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
\defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
\else
```

pdflatex only: Only pre-loaded if **pdflatex** is being used.

Pkg `microtype`

ligatures Older browsers don't display ligatures. Turn off letter ligatures, keeping LaTeX dash and quote ligatures, which may fail on older browers but at least won't corrupt written words.

```
\RequirePackage {microtype}

\microtypesetup{
    protrusion=false,
    expansion=false,
    tracking=false,
    kerning=false,
    spacing=false}

\DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

\fi

\end{warpHTML}
```

Pkg `geometry` Tactics to avoid unwanted page breaks and margin overflow:

- Uses a very long and wide page to minimize page breaks and margin overflow.
- Uses a scriptsize font.
- Uses extra space at the margin to avoid HTML tag overflow off the page.
- Forces a new PDF page before some environments.
- Forces line break between major pieces of long tags.

for HTML output:

```
\begin{warpHTML}
\RequirePackage[paperheight=190in,paperwidth=20in,%
left=2in,right=6in,%
top=1in,bottom=1in,%
]{geometry}
\@twosidefalse
\@mparswitchfalse
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Pkg `xparse`

LaTeX3 command argument parsing

```
\RequirePackage{xparse}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Pkg `expl3`

LaTeX3 programming

```
\RequirePackage{expl3}
```

Pkg `gettitlestring`

Used to emulate `\nameref`.

```
\RequirePackage{gettitlestring}
```

Pkg `everyhook`

everyhook is used to patch paragraph handling.

```
\RequirePackage{everyhook}
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Pkg filecontents

Used to write helper files, done in print mode.

Patched to work with **morewrites**, per https://tex.stackexchange.com/questions/312830/does-morewrites-not-support-filecontents-and-can-i-write-body-of-environment-us/312910

```
\RequirePackage{filecontents}

\@ifpackagelater{filecontents}{2011/10/09}%
{}
{
\newwrite\fcwrite
\let\LWR@origfilec@ntents\filec@ntents
\def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Pkg xifthen

```
\RequirePackage{xifthen}
```

Pkg xstring

```
\RequirePackage{xstring}
```

Pkg verbatim

```
\RequirePackage{verbatim}
```

Pkg calc

```
\RequirePackage{calc}
```

Pkg refcount

Provides \setcounterref, \setcounterpageref, etc.

```
\RequirePackage{refcount}
```

Pkg newfloat

```
\RequirePackage{newfloat}
```

```
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Pkg `environ` Used to encapsulate math environments for re-use in HTML `<alt>` text.

```
\RequirePackage{environ}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Pkg `zref` Used for cross-references.

```
\RequirePackage{zref}
```

Pkg `printlen` Used to convert lengths for image width/height options.

```
\RequirePackage{printlen}
```

`\LWR@printlength` {⟨*length*⟩}

Prints a length using a locally-controlled unit and space. Rounding is used unless the length is small.

```
\newrobustcmd*{\LWR@printlength}[1]{%
\begingroup%
\uselengthunit{PT}%
\renewcommand*{\unitspace}{}%
\ifdimless{#1}{10pt}{%
    \printlength{#1}%
}{%
    \rndprintlength{#1}%
}%
\endgroup%
}
```

```
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
```

Pkg `varwidth` Used for print-mode `lateximage`:

```
\RequirePackage{varwidth}
```

```
\end{warpprint}
```

27 Loading packages

for HTML output:

```
\begin{warpHTML}
```

Remember the original \RequirePackage:

```
\LetLtxMacro\LWR@origRequirePackage\RequirePackage
```

\LWR@requirepackagenames Stores the list of required package names.

```
\newcommand*{\LWR@requirepackagenames}{}
```

\LWR@parsedrequirepackagenames Stores the parsed list of required package names after spaces are removed and lwarp- is prepended.

```
\newcommand*{\LWR@parsedrequirepackagenames}{}
```

\LWR@findword [⟨*1: separator*⟩] {⟨*2: list*⟩} {⟨*3: index*⟩} [⟨*4: destination*⟩]

Note that argument 4 is passed directly to \StrBetween.

```
\newcommand*\LWR@findword[3][,]{%
    \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
}
```

\LWR@lookforpackagename {⟨*index*⟩}

If this is an **lwarp**-supported package name, re-direct it to the **lwarp** version by renaming it lwarp- followed by the original name.

Looks index deep into the list of package names, \LWR@requirepackagenames, and builds \LWR@parsedrequirepackagenames which is the modified list of names.

```
\newcommand*{\LWR@lookforpackagename}[1]{%
```

Find the index'th package name from the list:

```
\LWR@findword{\LWR@requirepackagenames}{#1}[\LWR@strresult]%
```

Remove blanks. The original name with blanks is in LWR@strresult and the final name with no blanks goes into LWR@strresulttwo.

```
\StrSubstitute[100]{\LWR@strresult}{ }{}[\LWR@strresulttwo]%
```

See if the package name was found:

```
\IfStrEq{\LWR@strresulttwo}{}%
{%
}% no filename
{% yes filename
```

If found, and if an **lwarp**-equivalent name exists, use `lwarp-*` instead.

```
    \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
    {% lwarp-* file found
        \ifdefvoid{\LWR@parsedrequirepackagenames}{%
            \edef\LWR@parsedrequirepackagenames{lwarp-\LWR@strresulttwo}%
        }{%
            \edef\LWR@parsedrequirepackagenames{%
                \LWR@parsedrequirepackagenames,lwarp-\LWR@strresulttwo%
            }%
        }%
    }%
    {%

        \ifdefvoid{\LWR@parsedrequirepackagenames}{%
            \edef\LWR@parsedrequirepackagenames{\LWR@strresulttwo}%
        }{%
            \edef\LWR@parsedrequirepackagenames{%
                \LWR@parsedrequirepackagenames,\LWR@strresulttwo%
            }%
        }%
    }% no lwarp-* file
}% yes filename
}
```

`\RequirePackage` `[`⟨*1: options*⟩`] {`⟨*2: package names*⟩`} [`⟨*3: version*⟩`]`

For each of many package names in a comma-separated list, if an **lwarp** version of a package exists, select it instead of the LaTeX version.

```
\RenewDocumentCommand{\RequirePackage}{o m o}{%
```

Redirect up to nine names:

```
\renewcommand*{\LWR@requirepackagenames}{#2}%
\renewcommand*{\LWR@parsedrequirepackagenames}{}%
\LWR@lookforpackagename{1}%
\LWR@lookforpackagename{2}%
```

```
\LWR@lookforpackagename{3}%
\LWR@lookforpackagename{4}%
\LWR@lookforpackagename{5}%
\LWR@lookforpackagename{6}%
\LWR@lookforpackagename{7}%
\LWR@lookforpackagename{8}%
\LWR@lookforpackagename{9}%
```

\RequirePackage depending on the options and version:

```
\IfValueTF{#1}%
{% options given
    \IfValueTF{#3}% version given?
    {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}[#3]}%
    {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}}%
}%
{% no options given
    \IfValueTF{#3}% version given?
    {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}[#3]}%
    {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}}%
}%
}
\LetLtxMacro\usepackage\RequirePackage
```

\LWR@ProvidesPackagePass {⟨*pkgname*⟩} [⟨*version*⟩]

Uses the original package, including options.

```
\NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
\PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
\IfValueTF{#2}
{\ProvidesPackage{lwarp-#1}[#2]}
{\ProvidesPackage{lwarp-#1}}
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
\ProcessOptions\relax
\IfValueTF{#2}
{\LWR@origRequirePackage{#1}[#2]}
{\LWR@origRequirePackage{#1}}
}
```

\LWR@ProvidesPackageDrop {⟨*pkgname*⟩} [⟨*version*⟩]

Ignores the original package and uses lwarp's version instead. Drops/discards all options.

```
\NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
\PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
\IfValueTF{#2}
```

```
{\ProvidesPackage{lwarp-#1}[#2]}
{\ProvidesPackage{lwarp-#1}}
```

Ignore all options.

```
\DeclareOption*{}
```

Nullifies then processes the options. Seems to be required when options contain curly braces, which were causing "`Missing \begin{document}`".

```
% \ProcessOptions\relax% original LaTeX code
\let\ds@\@empty%         from the original \ProcessOptions
\edef\@curroptions{}%    lwarp modification to \ProcessOptions
\@process@ptions\relax%  from the original \ProcessOptions
}
```

```
\end{warpHTML}
```

28 Additional required packages

for HTML output:

```
\begin{warpHTML}
```

Pkg `caption`

```
\RequirePackage{caption}%
```

```
\end{warpHTML}
```

29 File handles

Defines file handles for writes.

for HTML & PRINT:

```
\begin{warpall}
```

`\LWR@quickfile` For quick temporary use only. This is reused in several places.

```
\newwrite\LWR@quickfile%
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

\LWR@lateximagesfile For lateximages.txt.

```
\newwrite\LWR@lateximagesfile
```

```
\end{warpHTML}
```

30 Include a file

During HTML output, \include{<filename>} causes the following to occur:

1. **lwarp** creates <filename>_html_inc.tex whose contents are:
   \input <filename>.tex
2. <filename>_html_inc.tex is then \included instead of <filename>.tex.
3. <filename>_html_inc.aux is automatically generated and used by LaTeX.

for HTML output:

```
\begin{warpHTML}
```

\include {⟨*filename*⟩}

\@include {⟨*filename*⟩} Modified to load _html_inc files.

```
\def\@include#1 {%
\immediate\openout\LWR@quickfile #1_html_inc.tex% lwarp
\immediate\write\LWR@quickfile{\string\input{#1.tex}}% lwarp
\immediate\closeout\LWR@quickfile% lwarp
\LWR@origclearpage% \changed
\if@filesw
    \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
\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_html_inc.aux % changed
        \immediate\write\@partaux{\relax}%
```

```
    \fi
    \@input@{#1_html_inc.tex}% changed
    \LWR@origclearpage% changed
    \@writeckpt{#1}%
    \if@filesw
        \immediate\closeout\@partaux
    \fi
\else
    \deadcycles\z@
    \@nameuse{cp@#1}%
\fi
\let\@auxout\@mainaux%
}

\end{warpHTML}
```

31 Copying a file

for HTML output:

```
\begin{warpHTML}
```

`\LWR@copyfile` {⟨*source filename*⟩} {⟨*destination filename*⟩}

Used to copy the `.toc` file to `.sidetoc` to re-print the TOC in the sideTOC navigation pane.

```
\newwrite\LWR@copyoutfile  % open the file to write to
\newread\LWR@copyinfile      % open the file to read from

\newcommand*{\LWR@copyfile}[2]{%
\LWR@traceinfo{LWR@copyfile: copying #1 to #2}

\immediate\openout\LWR@copyoutfile=#2
\openin\LWR@copyinfile=#1
\begingroup\endlinechar=-1
\makeatletter

\LWR@traceinfo{LWR@copyfile: about to loop}

\loop\unless\ifeof\LWR@copyinfile
  \LWR@traceinfo{LWR@copyfile: one line}
  \read\LWR@copyinfile to\LWR@fileline  % Read one line and store it into \LWR@fileline
%   \LWR@fileline\par                   % print the content into the pdf
% print the content:
  \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
\repeat
\immediate\closeout\LWR@copyoutfile
```

```
\LWR@traceinfo{LWR@copyfile: done}
\endgroup
}

\end{warpHTML}
```

32 Debugging messages

for HTML & PRINT:

```
\begin{warpall}
```

Bool `LWR@tracinglwarp` True if tracing is turned on.

```
\newbool{LWR@tracinglwarp}
```

`\tracinglwarp` Turns on the debug tracing messages.

```
\newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}
```

`\LWR@traceinfo` `{`⟨*text*⟩`}` If tracing is turned on, writes the text to the `.log` file.

```
\newcommand{\LWR@traceinfo}[1]{%
\ifbool{LWR@tracinglwarp}%
{%
    \typeout{*** lwarp: #1}%
}%
{}%
}
```

Bool `HTMLDebugComments` Add comments in HTML about closing `<div>`s, sections, etc.

Default: `false`

```
\newbool{HTMLDebugComments}
\boolfalse{HTMLDebugComments}
```

If `\tracinglwarp`, show where preamble hooks occur:

```
\AfterEndPreamble{
\LWR@traceinfo{AfterEndPreamble}
}

\AtBeginDocument{
\LWR@traceinfo{AtBeginDocument}
}

\end{warpall}
```

33 Defining print and HTML versions of macros and environments

The following refers to defining objects inside **lwarp**, and is not for the user's document.

Many macros and environments must be provided as both print and HTML versions.

While generating the print version of a document, the original macros as defined by LaTeX and its packages are used as-is.

While generating the HTML version of a document, the original macro or environment is redefined to call a new HTML version or a copy of the original print version. The new HTML versions of macros and environments are used most of the time. Copies of the print versions are used inside a `lateximage` environment, which draws and remembers an image of the printed output, and also several other places. The copies of the print versions may also be used by the HTML versions, such as when the HTML version merely encloses the print version inside HTML tags.

The general structure for providing print and HTML versions of a macro or environment is as follows:

For a preexisting macro, not defined with xparse: An HTML version is provided with a special name, inside a `warpHTML` environment, then `\LWR@formatted` is used to redefine and patch various macros:

```
\begin{warpHTML}
\newcommand{\LWR@HTML@name}{...}% may also use xparse

\LWR@formatted{name}
\end{warpHTML}
```

`\LWR@formatted{name}` copies the original print version, then redefines `\name` to use either the print or HTML version depending on which mode **lwarp** is using. **xparse** may be used to define the new HTML version, even if the original did not use **xparse**.

For a preexisiting environment, not defined with xparse: The process is similar. Note the use of `\LWR@formattedenv` instead of `\LWR@formatted`.

```
\begin{warpHTML}
\newenvironment{LWR@HTML@name}{...}% may also use xparse
```

```
\LWR@formattedenv{name}
\end{warpHTML}
```

If the original used xparse: A copy must be made using a new name:

```
\begin{warpHTML}
\NewDocumentCommand{\LWR@print@name}{..}{..}% copy the original

\NewDocumentCommand{\LWR@HTML@name}{..}{..}% or use \newcommand

\LWR@formatted{name}
\end{warpHTML}
```

Similar for an environment, using `\LWR@formattedenv`. (`\LWR@formatted` and `\LWR@formattedenv` use `\LetLtxMacro` to copy the original print definiton, which may not work with macros and environments created by **xparse**, so the print version must be manually recreated in the **lwarp** source.)

For a new macro or environment, not using xparse for the print version:

```
\begin{warpall}
\newcommand{\name}{...}% NOT xparse!
\end{warpall}

\begin{warpHTML}
\newcommand{\LWR@HTML@name}{...}% may use xparse for HTML

\LWR@formatted{name}
\end{warpHTML}
```

Similar for an environment. The plain `\name` or environment `name` is used for the printed version, and is placed inside `warpall`. **xparse** may be used for the `\LWR@HTML@<name>` version.

For a new macro or environment, using xparse: It is possible to use **xparse** for an entirely new macro or environment by defining the `\LWR@print@<name>` version with **xparse**, along with `\name` defined without **xparse** to refer directly to the `\LWR@print` version:

```
\begin{warpall}
\NewDocumentCommand{\LWR@print@name}{...} {...}%      -or-
\NewDocumentEnvironment{\LWR@print@name}{...} {...} {...}

% Simply a call to \LWR@print@name:
\newcommand{\name}{\LWR@print@name}%        -or-
```

```
\newenvironment{name}{\LWR@print@name}{\endLWR@print@name}
\end{warpall}

\begin{warpHTML}
\NewDocumentCommand{\LWR@HTML@name}{...} {...}%      -or-
\NewDocumentEnvironment{\LWR@HTML@name}{...} {...} {...}

\LWR@formatted{name}%    -or-
\LWR@formattedenv{name}
\end{warpHTML}
```

In general, `\LWR@formatted` or `\LWR@formattedenv` are placed inside a `warpHTML` environment, and while producing an HTML document they do the following:

- Macros are modified:
  1. The pre-existing print version `\name` is saved as `\LWR@print@<name>`, unless `\LWR@print@<name>` is already defined.
  2. The original `\name` is redefined to call either the print or HTML version depending on which format is in use at the moment, as set by `\LWR@formatting`, which is defined as either "`print`" or "`HTML`".
- When **lwarp** is producing a print document, the original definitions are used, as well as any new definitions defined in `warpall` above.
- When **lwarp** is generating HTML output, `\LWR@formatting` is set to "`HTML`", and `\name` is directed to `\LWR@HTML@<name>`.
- When **lwarp** is generating HTML output but enters a `lateximage` environment, or for some other reason needs to draw images using the original print defintions, `\LWR@formatting` is changed to "`print`" and `\name` is then redirected to `\LWR@print@<name>`, which was the original `\name`.

Since arguments are not handled by the new `\name`, any star and other arguments are processed by the print or HTML version.

Expandable versions are also provided as well. These usually are necessary for anything which could appear inside a `tabular`, without which a "`Misplaced \omit`" error may occur.

⚠ Misplaced `\omit` error

```
\LWR@expandableformatted
\LWR@expandableformattedenv
```

(Older versions of **lwarp** used `\LetLtxMacro` for everything, but this could fail when using macros defined by **xparse**. This older system is still in use for many definitions.)

for HTML output: 844 `\begin{warpHTML}`

\LWR@formatting Remembers if selected print/HTML formatting.

Used while \LWR@restoreorigformatting, such as in an lateximage. May be set to either "print" or "HTML".

```
\newcommand*{\LWR@formatting}{HTML}
```

\LWR@formatted {⟨*macroname*⟩} No backslash in the macro name.

If not yet defined, defines \LWR@print@<name> as the original print-mode \<name>. Also redefines \<name> to use \LWR@<format>@<name>, where <format> is set by \LWR@formatting, and is print or HTML.

```
\newcommand*{\LWR@formatted}[1]{%
    \ifcsundef{LWR@print@#1}{%
        \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
            \csname#1\endcsname%
    }{}%
    \ifcsundef{#1}{%
        \expandafter\newrobustcmd\csname #1\endcsname{%
            \@nameuse{LWR@\LWR@formatting @#1}%
        }%
    }{%
        \expandafter\renewrobustcmd\csname #1\endcsname{%
            \@nameuse{LWR@\LWR@formatting @#1}%
        }%
    }%
}
```

\LWR@expandableformatted {⟨*macroname*⟩} No backslash in the macro name.

An expandable version of \LWR@formatted.

```
\newcommand*{\LWR@expandableformatted}[1]{%
    \ifcsundef{LWR@print@#1}{%
        \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
            \csname#1\endcsname%
    }{}%
    \ifcsundef{#1}{%
        \expandafter\newcommand\csname #1\endcsname{%
            \@nameuse{LWR@\LWR@formatting @#1}%
        }%
    }{%
        \expandafter\renewcommand\csname #1\endcsname{%
            \@nameuse{LWR@\LWR@formatting @#1}%
        }%
    }%
}
```

\LWR@formattedenv {⟨*environmentname*⟩}

If not yet defined, defines the environment LWR@print@<name> as the original print-mode <name>. Also redefines the environment <name> to use environment LWR@<format>@<name>, where <format> is set by \LWR@formatting, and is print or HTML.

```
\newcommand*{\LWR@formattedenv}[1]{%
    \ifcsundef{LWR@print@#1}{%
        \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
            \csname#1\endcsname%
        \csletcs{endLWR@print@#1}{end#1}%
    }{}%
    \DeclareDocumentEnvironment{#1}{}%
    {\@nameuse{LWR@\LWR@formatting @#1}}%
    {\@nameuse{endLWR@\LWR@formatting @#1}}%
}
```

\LWR@expandableformattedenv {⟨*environmentname*⟩}

An expandable version of LWR@formattedenv.

```
\newcommand*{\LWR@expandableformattedenv}[1]{%
    \ifcsundef{LWR@print@#1}{%
        \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
            \csname#1\endcsname%
        \csletcs{endLWR@print@#1}{end#1}%
    }{}%
    \DeclareExpandableDocumentEnvironment{#1}{}%
    {\@nameuse{LWR@\LWR@formatting @#1}}%
    {\@nameuse{endLWR@\LWR@formatting @#1}}%
}
```

```
\end{warpHTML}
```

34 HTML-conversion output modifications

These booleans modify the HTML output in various ways to improve conversion to EPUB or word processor imports.

for HTML & PRINT:

```
\begin{warpall}
```

34.1 User-level controls

Bool `FormatEPUB`
Default: `false`

Changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.

```
\newbool{FormatEPUB}
\boolfalse{FormatEPUB}
```

Bool `FormatWP`
Default: `false`

Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments.

```
\newbool{FormatWP}
\boolfalse{FormatWP}
```

Bool `WPMarkFloats`
Default: `false`

Adds

```
=== begin table ===
...
=== end ===
```

or

```
=== begin figure ===
...
=== end ===
```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames and captions. [16]

```
\newbool{WPMarkFloats}
\boolfalse{WPMarkFloats}
```

Bool `WPMarkMinipages`
Default: `false`

Adds

```
=== begin minipage ===
...
=== end minipage ===
```

around minipages while formatting for word processors. This helps identify boundaries of minipages to be manually converted to word-processor frames.

```
\newbool{WPMarkMinipages}
\boolfalse{WPMarkMinipages}
```

Bool `WPMarkTOC`
Default: `true`

While formatting for word processors, adds

[16] Perhaps some day word processors will have HTML import options for identifying `<figure>` and `<figcaption>` tags for figures and tables.

```
=== table of contents ===
```

where the Table of Contents would have been. This helps identify where to insert the actual TOC.

If set `false`, the actual TOC is printed instead.

```
\newbool{WPMarkTOC}
\booltrue{WPMarkTOC}
```

Bool `WPMarkLOFT`
Default: `false`

While formatting for word processors, adds

```
=== list of figures ===    and/or
=== list of tables ===
```

where each of these lists would have been. This helps identify where to insert the actual lists.

If set `false`, the actual lists are printed instead.

```
\newbool{WPMarkLOFT}
\boolfalse{WPMarkLOFT}
```

Bool `WPMarkMath`
Default: `false`

While formatting for word processors, prints math as LaTeX code instead of creating SVG images or MathJax. This is useful for cut/paste into the **LibreOffice Writer TeXMaths** extension.

```
\newbool{WPMarkMath}
\boolfalse{WPMarkMath}
```

Bool `WPTitleHeading`
Default: `false`

While formatting for word processors, `true` sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to `false` to cause the title to be plain text, and the section headings to begin at **Heading 1**.

See table 6 on page 169.

```
\newbool{WPTitleHeading}
\boolfalse{WPTitleHeading}
```

```
\end{warpall}
```

34.2 Heading adjustments

If formatting the HTML for a word processor, adjust heading levels.

If WPTitleHeading is true, adjust so that part is **Heading 1**.

If WPTitleHeading is false, use <h1> for the title, and set part to **Heading 2**.

for HTML output:

```
\begin{warpHTML}

\AtBeginDocument{
\ifbool{FormatWP}{
\@ifundefined{chapter}{
\ifbool{WPTitleHeading}{% part and section starting at h2
\renewcommand*{\LWR@tagtitle}{h1}
\renewcommand*{\LWR@tagtitleend}{/h1}
\renewcommand*{\LWR@tagpart}{h2}
\renewcommand*{\LWR@tagpartend}{/h2}
\renewcommand*{\LWR@tagsection}{h3}
\renewcommand*{\LWR@tagsectionend}{/h3}
\renewcommand*{\LWR@tagsubsection}{h4}
\renewcommand*{\LWR@tagsubsectionend}{/h4}
\renewcommand*{\LWR@tagsubsubsection}{h5}
\renewcommand*{\LWR@tagsubsubsectionend}{/h5}
\renewcommand*{\LWR@tagparagraph}{h6}
\renewcommand*{\LWR@tagparagraphend}{/h6}
\renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
\renewcommand*{\LWR@tagsubparagraphend}{/span}
}% WPTitleHeading
{% not WPTitleHeading, part and section starting at h1
\renewcommand*{\LWR@tagtitle}{div class="title"}
\renewcommand*{\LWR@tagtitleend}{/div}
\renewcommand*{\LWR@tagpart}{h1}
\renewcommand*{\LWR@tagpartend}{/h1}
\renewcommand*{\LWR@tagsection}{h2}
\renewcommand*{\LWR@tagsectionend}{/h2}
\renewcommand*{\LWR@tagsubsection}{h3}
\renewcommand*{\LWR@tagsubsectionend}{/h3}
\renewcommand*{\LWR@tagsubsubsection}{h4}
\renewcommand*{\LWR@tagsubsubsectionend}{/h4}
\renewcommand*{\LWR@tagparagraph}{h5}
\renewcommand*{\LWR@tagparagraphend}{/h5}
\renewcommand*{\LWR@tagsubparagraph}{h6}
\renewcommand*{\LWR@tagsubparagraphend}{/h6}
}% not WPTitleHeading
}% chapter undefined
{% chapter defined
\ifbool{WPTitleHeading}{}
{% not WPTitleHeading, part and chapter starting at h1
\renewcommand*{\LWR@tagtitle}{div class="title"}
\renewcommand*{\LWR@tagtitleend}{/div}
\renewcommand*{\LWR@tagpart}{h1}
\renewcommand*{\LWR@tagpartend}{/h1}
\renewcommand*{\LWR@tagchapter}{h2}
```

```
\renewcommand*{\LWR@tagchapterend}{/h2}
\renewcommand*{\LWR@tagsection}{h3}
\renewcommand*{\LWR@tagsectionend}{/h3}
\renewcommand*{\LWR@tagsubsection}{h4}
\renewcommand*{\LWR@tagsubsectionend}{/h4}
\renewcommand*{\LWR@tagsubsubsection}{h5}
\renewcommand*{\LWR@tagsubsubsectionend}{/h5}
\renewcommand*{\LWR@tagparagraph}{h6}
\renewcommand*{\LWR@tagparagraphend}{/h6}
\renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
\renewcommand*{\LWR@tagsubparagraphend}{/span}
}% not WPTitleHeading
}% chapter defined
}{}% FormatWP
}% AtBeginDocument

\end{warpHTML}
```

35 Remembering original formatting macros

for HTML output:

```
\begin{warpHTML}
```

Remember original definitions of formatting commands. Will be changed to HTML commands for most uses. Will be temporarily restored to original meaning inside any lateximage environment. Also nullify unused commands.

Some packages redefine \#, which is used to generate HTML, so the original must be remembered here.

```
\chardef\LWR@origpound=`\#

\let\LWR@origcomma\,
\let\LWR@origthinspace\thinspace
\let\LWR@origtilde~
\let\LWR@origenskip\enskip
\let\LWR@origquad\quad
\let\LWR@origqquad\qquad
\let\LWR@orighfil\hfil
\let\LWR@orighss\hss
\let\LWR@origllap\llap
\let\LWR@origrlap\rlap
\let\LWR@orighfilneg\hfilneg

\let\LWR@origmedskip\medskip
\let\LWR@origbigskip\bigskip

```

```
\let\LWR@origtextellipsis\textellipsis

\LetLtxMacro\LWR@origtextrm\textrm
\LetLtxMacro\LWR@origtextsf\textsf
\LetLtxMacro\LWR@origtexttt\texttt
\LetLtxMacro\LWR@origtextnormal\textnormal
\LetLtxMacro\LWR@origtextbf\textbf
\LetLtxMacro\LWR@origtextmd\textmd
\LetLtxMacro\LWR@origtextit\textit
\LetLtxMacro\LWR@origtextsl\textsl
\LetLtxMacro\LWR@origtextsc\textsc
\LetLtxMacro\LWR@origtextup\textup
\LetLtxMacro\LWR@origemph\emph

\LetLtxMacro\LWR@origrmfamily\rmfamily
\LetLtxMacro\LWR@origsffamily\sffamily
\LetLtxMacro\LWR@origttfamily\ttfamily
\LetLtxMacro\LWR@origbfseries\bfseries
\LetLtxMacro\LWR@origmdseries\mdseries
\LetLtxMacro\LWR@origupshape\upshape
\LetLtxMacro\LWR@origslshape\slshape
\LetLtxMacro\LWR@origscshape\scshape
\LetLtxMacro\LWR@origitshape\itshape
\LetLtxMacro\LWR@origem\em
\LetLtxMacro\LWR@orignormalfont\normalfont

\let\LWR@origonecolumn\onecolumn

\let\LWR@origsp\sp
\let\LWR@origsb\sb
\LetLtxMacro\LWR@origtextsuperscript\textsuperscript
\LetLtxMacro\LWR@orig@textsuperscript\@textsuperscript

\AtBeginDocument{
\LetLtxMacro\LWR@origtextsubscript\textsubscript
\LetLtxMacro\LWR@orig@textsubscript\@textsubscript
}

\LetLtxMacro\LWR@origunderline\underline


\let\LWR@orignewpage\newpage

\let\LWR@origpagestyle\pagestyle
\let\LWR@origthispagestyle\thispagestyle
\LetLtxMacro\LWR@origpagenumbering\pagenumbering

\let\LWR@orignewline\newline


```

```
\AtBeginDocument{% in case packages change definition
\let\LWR@orig@trivlist\@trivlist
\let\LWR@origtrivlist\trivlist
\let\LWR@origendtrivlist\endtrivlist
\LetLtxMacro\LWR@origitem\item
\LetLtxMacro\LWR@origitemize\itemize
\LetLtxMacro\LWR@endorigitemize\enditemize
\LetLtxMacro\LWR@origenumerate\enumerate
\LetLtxMacro\LWR@endorigenumerate\endenumerate
\LetLtxMacro\LWR@origdescription\description
\LetLtxMacro\LWR@endorigdescription\enddescription
\let\LWR@orig@mklab\@mklab
\let\LWR@origmakelabel\makelabel
\let\LWR@orig@donoparitem\@donoparitem
\LetLtxMacro\LWR@orig@item\@item
\let\LWR@orig@nbitem\@nbitem
}

\let\LWR@origpar\par

\LetLtxMacro\LWR@origfootnote\footnote
\let\LWR@orig@mpfootnotetext\@mpfootnotetext

\let\LWR@origclearpage\clearpage


\AtBeginDocument{% in case packages change definition
\LetLtxMacro\LWR@orighline\hline%
\LetLtxMacro\LWR@origcline\cline%
}

\end{warpHTML}
```

36 Accents

Native LaTeX accents such as `\"` will work, but many more kinds of accents are available when using Unicode-aware XeLaTeX and LuaLaTeX.

for HTML output:

```
\begin{warpHTML}
```

Without `\AtBeginDocument`, `\t` was being re-defined somewhere.

```
\AtBeginDocument{
```

The following are restored for print when inside a `lateximage`.

For Unicode engines, only \t needs to be redefined:

```
\LetLtxMacro\LWR@origt\t
```

For pdfLaTeX, additional work is required:

```
\ifPDFTeX
\LetLtxMacro\LWR@origequalaccent\=
\LetLtxMacro\LWR@origdotaccent\.
\LetLtxMacro\LWR@origu\u
\LetLtxMacro\LWR@origv\v
\LetLtxMacro\LWR@origc\c
\LetLtxMacro\LWR@origd\d
\LetLtxMacro\LWR@origb\b
```

The HTML redefinitions follow.

For pdfLaTeX, Unicode diacritical marks are used:

```
\renewcommand*{\=}[1]{#1\HTMLunicode{0305}}
\renewcommand*{\.}[1]{#1\HTMLunicode{0307}}
\renewcommand*{\u}[1]{#1\HTMLunicode{0306}}
\renewcommand*{\v}[1]{#1\HTMLunicode{030C}}
\renewcommand*{\c}[1]{#1\HTMLunicode{0327}}
\renewcommand*{\d}[1]{#1\HTMLunicode{0323}}
\renewcommand*{\b}[1]{#1\HTMLunicode{0331}}
\fi
```

For all engines, a Unicode diacritical tie is used:

```
\def\LWR@t#1#2{#1\HTMLunicode{0361}#2}
\renewcommand*{\t}[1]{\LWR@t#1}
```

\LWR@restoreorigaccents Called from \restoreoriginalformatting when a lateximage is begun.

```
\ifPDFTeX
\newcommand*{\LWR@restoreorigaccents}{%
\LetLtxMacro\=\LWR@origequalaccent%
\LetLtxMacro\.\LWR@origdotaccent%
\LetLtxMacro\u\LWR@origu%
\LetLtxMacro\v\LWR@origu%
\LetLtxMacro\t\LWR@origt%
\LetLtxMacro\c\LWR@origc%
\LetLtxMacro\d\LWR@origd%
\LetLtxMacro\b\LWR@origb%
}%
\else% XeLaTeX, LuaLaTeX:
\newcommand*{\LWR@restoreorigaccents}{%
```

```
\LetLtxMacro\t\LWR@origt%
}%
\fi%
}% AtBeginDocument

\end{warpHTML}
```

37 Configuration Files

```
\begin{warpprint}
\typeout{lwarp: generating configuration files}
\end{warpprint}
```

37.1 project_html.tex

File `project_html.tex` Used to allow an HTML version of the document to exist alongside the print version.

Only write `\jobname_html.tex` if generating the print version.

```
\begin{warpprint}
\immediate\openout\LWR@quickfile=\jobname_html.tex
\immediate\write\LWR@quickfile{%
\detokenize{\PassOptionsToPackage}%
{warpHTML,BaseJobname=\jobname}{lwarp}%
}
\immediate\write\LWR@quickfile{%
\detokenize{\input}\string{\jobname.tex\string }%
}
\immediate\closeout\LWR@quickfile
\end{warpprint}
```

37.2 lwarpmk configuration files

for PRINT output:

```
\begin{warpprint}
```

`\LWR@lwarpconfversion` The version number of the configuration file, allowing **lwarpmk** to detect an obsolete configuration file format. Incremented by one each time the configuration file format changes. (This is NOT the same as the **lwarp** version number.)

```
\newcommand*{\LWR@lwarpconfversion}{1}% also in lwarpmk.lua
```

The literal dollar sign character, with a plain catcode, allowing it to be written to a file as-is.

```
\catcode`\$=12
\def\LWR@dollarchar{$}
\catcode`\$=3
```

The literal ampersand character, with a plain catcode, allowing it to be written to a file as-is.

```
\catcode`\&=12
\def\LWR@ampersand{&}
\catcode`\&=4
```

`\LWR@opquote` The operating system's quote mark:

```
\ifbool{usingOSWindows}{
    \def\LWR@opquote{"}
}{
    \def\LWR@opquote{'}
}
```

`\LWR@opsequence` The operating system's sequential execution command:

```
\ifbool{usingOSWindows}{
    \def\LWR@opsequence{\LWR@ampersand}
}{
    \def\LWR@opsequence{\LWR@ampersand\LWR@ampersand}
}
```

`\LWR@shellescapecmd` The LaTeX compile option for shell escape, if used.

```
\ifshellescape
    \def\LWR@shellescapecmd{--shell-escape }
\else
    \def\LWR@shellescapecmd{}
\fi
```

`\LWR@compilecmd` {⟨*engine*⟩} {⟨*suffix*⟩}

Used to form the basic compilation command for a document, adding the optional shell escape.

Engine is **pdflatex**, etc. Suffix is empty or `_html`

```
\newcommand*{\LWR@compilecmd}[2]{%
```

```
    #1 \LWR@shellescapecmd \jobname#2%
}
```

`\LWR@addcompilecmd` {⟨*cmd*⟩} {⟨*suffix*⟩}

Adds to the compilation command.

Cmd is **dvipdfmx**, etc. Suffix is empty or `_html`

```
\newcommand*{\LWR@addcompilecmd}[2]{%
    \space \LWR@opsequence \space\space
    #1 \jobname#2%
}
```

`\LWR@unknownengine` Error message if not sure which LaTeX engine is being used.

```
\newcommand*{\LWR@unknownengine}{%
    \PackageError{lwarp}
    {Unknown LaTeX engine.}
    {Lwarp only knows about pdflatex, dvi latex, xelatex, and lualatex.}
}
```

`\LWR@latexmkvar` {⟨*varname*⟩} {⟨*value*⟩}

Adds a **latexmk** variable assignment.

```
\newcommand*{\LWR@latexmkvar}[2]{%
    -e
    \LWR@opquote%
    \LWR@dollarchar #1=q/#2/%
    \LWR@opquote
}
```

`\LWR@latexmkcmd` {⟨***latexmk** options*⟩}

Sets a call to **latexmk** with the given options, possibly adding `--shell-escape`, and also adding the indexing program.

```
\newcommand*{\LWR@latexmkcmd}[1]{%
    latexmk  \space \LWR@shellescapecmd \space  #1 \space
    -recorder \space
    \LWR@latexmkvar{makeindex}{\LWR@LatexmkIndexCmd}
}
```

`\LWR@latexmkdvipdfm` {⟨***dvipdfm** or **dvipdfmx***⟩}

Adds the options settings for **dvipdfm** or **dvipdfmx**.

```
\newcommand*{\LWR@latexmkdvipdfm}[1]{%
    -pdfdvi \space
    \LWR@latexmkvar{dvipdf}{%
        #1
        \@percentchar O
        -o \@percentchar D
        \@percentchar S%
    }
}
```

\LWR@printlatexcmd
\LWR@HTMLlatexcmd

The following sets the command to use to compile the source to PDF form.

If using **latexmk**, a complicated string is created, eventually resulting in something such as:

For **xelatex** with `-shell-escape`:

```
[[latexmk   -xelatex   --shell-escape  -recorder
    -e '$makeindex = q/makeindex -s lwarp.ist/'  <jobname>_html]]
```

For **dvipdfmx**:

```
[[latexmk   -pdfdvi  -e '$dvipdf=q/dvipdfmx %O -o %D %S/'
    -recorder
    -e '$makeindex=q/makeindex -s lwarp.ist/'   <jobname>_html]]
```

```
\ifbool{LWR@latexmk}{
```

For **latexmk** with **pdflatex** or **lualatex**:

```
    \ifpdf
```

For **latexmk** with **pdflatex**:

```
        \ifPDFTeX
            \def\LWR@latexcmd{\LWR@latexmkcmd{-pdf -dvi- -ps-}}
        \else
```

For **latexmk** with **lualatex**:

```
            \ifLuaTeX
```

```
                \def\LWR@latexcmd{\LWR@latexmkcmd{-lualatex}}
            \else
                \LWR@unknownengine
            \fi
        \fi
    \else% \ifpdf
```

For **latexmk** with **xelatex** or DVI **latex**:

```
        \ifXeTeX
```

For **latexmk** with **xelatex**:

```
            \def\LWR@latexcmd{\LWR@latexmkcmd{-xelatex}}
        \else% \ifXeTeX
```

For **latexmk** with DVI **latex**:

```
            \ifbool{LWR@dvipdfm}{
                \def\LWR@latexcmd{%
                    \LWR@latexmkcmd{%
                        \LWR@latexmkdvipdfm{dvipdfm}%
                    }
                }
            }{
                \ifbool{LWR@dvipdfmx}{
                    \def\LWR@latexcmd{%
                        \LWR@latexmkcmd{%
                            \LWR@latexmkdvipdfm{dvipdfmx}%
                        }
                    }
                }{
                    \def\LWR@latexcmd{\LWR@latexmkcmd{-pdfps}}
                }
            }
        \fi
    \fi% \ifpdf
```

The final assignment if **latexmk**:

```
    \def\LWR@printlatexcmd{\LWR@latexcmd \space \jobname}
    \def\LWR@HTMLlatexcmd{\LWR@latexcmd \space \jobname_html}
}% latexmk
```

Without **latexmk**, the compiling command is simply the compiler name and the optional shell escape:

```
{% not latexmk
    \ifpdf
```

For **pdflatex** or **lualatex**:

```
        \ifPDFTeX
```

For **pdflatex**:

```
            \def\LWR@printlatexcmd{\LWR@compilecmd{pdflatex}{}}
            \def\LWR@HTMLlatexcmd{\LWR@compilecmd{pdflatex}{_html}}
        \else
            \ifLuaTeX
```

For **lualatex**:

```
                \def\LWR@printlatexcmd{\LWR@compilecmd{lualatex}{}}
                \def\LWR@HTMLlatexcmd{\LWR@compilecmd{lualatex}{_html}}
            \else
                \LWR@unknownengine
            \fi
        \fi
    \else% \ifpdf
```

For DVI **latex** or **xelatex**:

```
        \ifXeTeX
```

For **xelatex**:

```
            \def\LWR@printlatexcmd{\LWR@compilecmd{xelatex}{}}
            \def\LWR@HTMLlatexcmd{\LWR@compilecmd{xelatex}{_html}}
        \else
```

For DVI **latex**. Default to **dvips**, unless told to use **dvipdfm** or **dvipdfmx**:

```
            \ifbool{LWR@dvipdfm}{
```

For DVI **latex** with **dvipdfm**:

```
                \def\LWR@printlatexcmd{%
                    \LWR@compilecmd{latex}{}
                    \LWR@addcompilecmd{dvipdfm}{}
                }
                \def\LWR@HTMLlatexcmd{%
                    \LWR@compilecmd{latex}{_html}
                    \LWR@addcompilecmd{dvipdfm}{_html}
                }
            }{
                \ifbool{LWR@dvipdfmx}{
```

For DVI **latex** with **dvipdfmx**:

```
                    \def\LWR@printlatexcmd{%
                        \LWR@compilecmd{latex}{}
                        \LWR@addcompilecmd{dvipdfmx}{}
                    }
                    \def\LWR@HTMLlatexcmd{%
                        \LWR@compilecmd{latex}{_html}
                        \LWR@addcompilecmd{dvipdfmx}{_html}
                    }
                }{% dvips
```

For DVI **latex** with **dvips** and **ps2pdf**:

```
                    \def\LWR@printlatexcmd{%
                        \LWR@compilecmd{latex}{}
                        \LWR@addcompilecmd{dvips}{}
                        \LWR@addcompilecmd{ps2pdf}{}.ps
                    }
                    \def\LWR@HTMLlatexcmd{%
                        \LWR@compilecmd{latex}{_html}
                        \LWR@addcompilecmd{dvips}{_html}
                        \LWR@addcompilecmd{ps2pdf}{_html}.ps
                    }
                }
            }
        \fi% \ifXeTeX
    \fi% \ifpdf
}% latexmk
```

`\LWR@writeconf` {⟨*filename*⟩}

Common code for each of `lwarpmk.conf` and `<project>.lwarpmkconf`. Each entry is a variable name, the equal sign, and a quoted string inside `[[` and `]]`, which are **lua**'s long quote characters, allowing the use of single and double quotes inside.

```
\newcommand{\LWR@writeconf}[1]{%
\ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
\immediate\openout\LWR@quickfile=#1
\immediate\write\LWR@quickfile{confversion = [[\LWR@lwarpconfversion]]}
\ifbool{usingOSWindows}{
    \immediate\write\LWR@quickfile{opsystem = [[Windows]]}
}{
    \immediate\write\LWR@quickfile{opsystem = [[Unix]]}
}
\immediate\write\LWR@quickfile{sourcename = [[\jobname]]}
\immediate\write\LWR@quickfile{homehtmlfilename = [[\HomeHTMLFilename]]}
\immediate\write\LWR@quickfile{htmlfilename = [[\HTMLFilename]]}
```

```
\immediate\write\LWR@quickfile{latexmk = [[\ifbool{LWR@latexmk}{true}{false}]]}
\immediate\write\LWR@quickfile{printlatexcmd = [[\LWR@printlatexcmd]]}
\immediate\write\LWR@quickfile{HTMLlatexcmd = [[\LWR@HTMLlatexcmd]]}
\immediate\write\LWR@quickfile{printindexcmd = [[\LWR@PrintIndexCmd]]}
\immediate\write\LWR@quickfile{HTMLindexcmd = [[\LWR@HTMLIndexCmd]]}
\immediate\write\LWR@quickfile{latexmkindexcmd = [[\LWR@LatexmkIndexCmd]]}
\immediate\write\LWR@quickfile{glossarycmd = [[\LWR@GlossaryCmd]]}
\immediate\write\LWR@quickfile{pdftotextenc = [[\LWR@pdftotextEnc]]}
\immediate\closeout\LWR@quickfile
}

\end{warpprint}
```

37.2.1 lwarpmk.conf

File `lwarpmk.conf` `lwarpmk.conf` is automatically (re-)created by the **lwarp** package when executing
`pdflatex <project.tex>`,
or similar for **xelatex** or **lualatex**, in print-document generation mode, which is the default unless the `warpHTML` option is given. `lwarpmk.conf` is then used by the utility **lwarpmk**.

for PRINT output:

```
\begin{warpprint}

\AtBeginDocument{\LWR@writeconf{lwarpmk.conf}}

\end{warpprint}
```

37.2.2 <project>.lwarpmkconf

File `project.lwarpmkconf` A project-specific configuration file for **lwarpmk**.

The `makeindex` and `xindy` options have already been handled for `lwarp.conf`.

```
\begin{warpprint}

\AtBeginDocument{\LWR@writeconf{\jobname.lwarpmkconf}}

\end{warpprint}
```

37.3 lwarp.css

File `lwarp.css` This is the base css layer used by **lwarp**.

This must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp.css}
/*
  CSS stylesheet for the LaTeX lwarp package
  Copyright 2016-2018 Brian Dunn -- BD Tech Concepts LLC
*/


/* a fix for older browsers: */
header, section, footer, aside, nav, main,
    article, figure { display: block; }


A:link {color:#000080 ; text-decoration: none ; }
A:visited {color:#800000 ; }
A:hover {color:#000080 ; text-decoration: underline ;}
A:active {color:#800000 ; }

a.tocpart {display: inline-block ; margin-left: 0em ;
    font-weight: bold ;}
a.tocchapter {display: inline-block ; margin-left: 0em ;
    font-weight: bold ;}
a.tocsection {display: inline-block ; margin-left: 1em ;
    text-indent: -.5em ; font-weight: bold ; }
a.tocsubsection {display: inline-block ; margin-left: 2em ;
    text-indent: -.5em ; }
a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
    text-indent: -.5em ; }
a.tocparagraph {display: inline-block ; margin-left: 4em ;
    text-indent: -.5em ; }
a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
    text-indent: -.5em ; }
a.tocfigure {margin-left: 0em}
a.tocsubfigure {margin-left: 2em}
a.toctable {margin-left: 0em}
a.tocsubtable {margin-left: 2em}
a.toctheorem {margin-left: 0em}
a.toclstlisting {margin-left: 0em}

body {
    font-family: "DejaVu Serif", "Bitstream Vera Serif",
        "Lucida Bright", Georgia, serif;
    background: #FAF7F4 ;
    color: black ;
    margin:0em ;
    padding:0em ;
```

```
1350     font-size: 100% ;
1351     line-height: 1.2 ;
1352 }
1353
1354 p {margin: 1.5ex 0em 1.5ex 0em ;}
1355 table p {margin: .5ex 0em .5ex 0em ;}
1356
1357 /* Holds a section number to add space between it and the name */
1358 span.sectionnumber { margin-right: 0em }
1359
1360 /* Inserted in front of index lines */
1361 span.indexitem {margin-left: 0em}
1362 span.indexsubitem {margin-left: 2em}
1363 span.indexsubsubitem {margin-left: 4em}
1364
1365 div.hidden, span.hidden { display: none ; }
1366
1367 kbd, span.texttt {
1368     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1369         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1370         "Courier New", monospace;
1371     font-size: 100% ;
1372 }
1373
1374 pre { padding: 3pt ; }
1375
1376 span.strong, span.textbf, div.strong, div.textbf { font-weight: bold; }
1377
1378 span.textit, div.textit { font-style: italic; }
1379
1380 span.textmd, div.textmd { font-weight: normal; }
1381
1382 span.textsc, div.textsc { font-variant: small-caps; }
1383
1384 span.textsl, div.textsl { font-style: oblique; }
1385
1386 span.textup, div.textup { font-variant: normal; }
1387
1388 span.textrm, div.textrm {
1389     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1390     "Lucida Bright", Georgia, serif;
1391 }
1392
1393 span.textsf, div.textsf {
1394      font-family: "DejaVu Sans", "Bitstream Vera Sans",
1395         Geneva, Verdana, sans-serif ;
1396 }
1397
1398 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
1399
```

```
span.underline {
    text-decoration: underline ;
    text-decoration-skip: auto ;
}

span.overline {
    text-decoration: overline ;
    text-decoration-skip: auto ;
}

/* for diagbox */
div.diagboxtitleN { border-bottom: 1px solid gray }
div.diagboxtitleS { border-top: 1px solid gray }

div.diagboxE {
    padding-left: 2em ;
    text-align: right ;
}

div.diagboxW {
    padding-right: 2em ;
    text-align: left ;
}



/* For realscripts */
.supsubscript {
    display: inline-block;
    text-align:left ;
}

.supsubscript sup,
.supsubscript sub {
    position: relative;
    display: block;
    font-size: .5em;
    line-height: 1;
}

.supsubscript sup {
    top: .5em;
}

.supsubscript sub {
    top: .5em;
}

div.attribution p {
    text-align: right ;
```

```
1450     font-size: 80%
1451 }
1452
1453 span.poemtitle {
1454   font-size: 120% ; font-weight: bold;
1455 }
1456
1457 pre.tabbing {
1458     font-family: "Linux Libertine Mono O", "Lucida Console",
1459         "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1460         "Liberation Mono", "FreeMono", "Andale Mono",
1461         "Nimbus Mono L", "Courier New", monospace;
1462 }
1463
1464 blockquote {
1465     margin-left: 0px ;
1466     margin-right: 0px ;
1467 }
1468
1469 /* quotchap is for the quotchap package */
1470 div.quotchap {
1471   font-style: oblique ;
1472   overflow-x: auto ;
1473   margin-left: 2em ;
1474   margin-right: 2em ;
1475 }
1476
1477 blockquote p, div.quotchap p {
1478   line-height: 1.5;
1479     text-align: left ;
1480     font-size: .85em ;
1481     margin-left: 3em ;
1482     margin-right: 3em ;
1483 }
1484
1485 /* qauthor is for the quotchap package */
1486 div.qauthor {
1487   display: block ;
1488   text-align: right ;
1489   margin-left: auto ;
1490   margin-right: 2em ;
1491   font-size: 80% ;
1492   font-variant: small-caps;
1493 }
1494
1495 div.qauthor p {
1496   text-align: right ;
1497 }
1498
1499 blockquotation {
```

```
    margin-left: 0px ;
    margin-right: 0px ;
}

blockquotation p {
  line-height: 1.5;
    text-align: left ;
    font-size: .85em ;
    margin-left: 3em ;
    margin-right: 3em ;
}

div.epigraph, div.dictum {
  line-height: 1.2;
    text-align: left ;
    padding: 3ex 1em 0ex 1em ;
/*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
    margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
/*    margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
    font-size: .85em ;
    max-width: 27em ;
}



div.epigraphsource, div.dictumauthor {
    text-align:right ;
    margin-left:auto ;
/*     max-width: 50% ; */
    border-top: 1px solid #A0A0A0 ;
    padding-bottom: 3ex ;
    line-height: 1.2;
}

div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
div.dictumauthor { font-style:italic }


/* lettrine package: */
span.lettrine { font-size: 4ex ; float: left ; }
span.lettrinetext { font-variant: small-caps ; }

/* ulem and soul packages: */
span.uline {
    text-decoration: underline ;
    text-decoration-skip: auto ;
}

span.uuline {
```

```
    text-decoration: underline ;
    text-decoration-skip: auto ;
    text-decoration-style: double ;
}

span.uwave {
    text-decoration: underline ;
    text-decoration-skip: auto ;
    text-decoration-style: wavy ;
}

span.sout {
    text-decoration: line-through ;
}

span.xout {
    text-decoration: line-through ;
}

span.dashuline {
    text-decoration: underline ;
    text-decoration-skip: auto ;
    text-decoration-style: dashed ;
}

span.dotuline {
    text-decoration: underline ;
    text-decoration-skip: auto ;
    text-decoration-style: dotted ;
}

span.letterspacing { letter-spacing: .2ex ; }

span.capsspacing {
    font-variant: small-caps ;
    letter-spacing: .1ex ;
}

span.highlight { background: #F8E800 ; }



html body {
  margin: 0 ;
  line-height: 1.2;
}


body div {
```

```
  margin: 0ex;
}


h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    font-style: normal ;
    font-weight: bold ;
    text-align: left ;
}

h1 {    /* title of the entire website, used on each page */
    text-align: center ;
    font-size: 2.5em ;
    padding: .4ex 0em 0ex 0em ;
}
h2 { font-size: 2.25em }
h3 { font-size: 2em }
h4 { font-size: 1.75em }
h5 { font-size: 1.5em }
h6 { font-size: 1.25em }
span.paragraph {font-size: 1em ; font-variant: normal ;
    margin-right: 1em ; }
span.subparagraph {font-size: 1em ; font-variant: normal ;
    margin-right: 1em ;  }

div.minisec {
    font-family: "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
    font-style: normal ;
    font-weight: bold ;
    text-align: left ;
}

/* Title of the file */
h1 {
  margin: 0ex 0em 0ex 0em ;
  line-height: 1.3;
  text-align: center ;
}

/* Part */
h2 {
  margin: 1ex 0em 1ex 0em ;
  line-height: 1.3;
  text-align: center ;
```

```
}

/* Chapter */
h3 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Section */
h4 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Sub-Section */
h5 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}

/* Sub-Sub-Section */
h6 {
  margin: 3ex 0em 1ex 0em ;
  line-height: 1.3;
}


div.titlepage {
  text-align: center ;
}

.footnotes {
    font-size: .85em ;
    margin: 3ex 2em 0ex 2em ;
    border-top: 1px solid silver ;
}

.marginpar, .marginparblock {
    max-width:50%;
    float:right;
    text-align:left;
    margin: 1ex 0.5em 1ex 1em ;
    padding: 1ex 0.5em 1ex 0.5em ;
    font-size: 85% ;
    border-top: 1px solid silver ;
    border-bottom: 1px solid silver ;
    overflow-x: auto;
}

.marginpar br { margin-bottom: 2ex ; }
```

```
1700
1701 div.marginblock, div.marginparblock {
1702     max-width:50%;
1703     float:right;
1704     text-align:left;
1705     margin: 1ex 0.5em 1ex 1em ;
1706     padding: 1ex 0.5em 1ex 0.5em ;
1707     overflow-x: auto;
1708 }
1709
1710 div.marginblock div.minipage,
1711 div.marginparblock div.minipage {
1712     display: block ;
1713     margin: 0pt auto 0pt auto ;
1714 }
1715
1716 div.marginblock div.minipage p ,
1717 div.marginparblock div.minipage p
1718     { font-size: 85%}
1719
1720 div.marginblock br ,
1721 div.marginparblock br
1722     { margin-bottom: 2ex ; }
1723
1724
1725 section.textbody div.footnotes{
1726     margin: 3ex 2em 0ex 2em ;
1727     border-bottom: 2px solid silver ;
1728 }
1729
1730 .footnoteheader {
1731     border-top: 2px solid silver ;
1732     margin-top: 3ex ;
1733     padding-top: 1ex ;
1734     font-weight: bold ;
1735 }
1736
1737 .mpfootnotes {
1738     text-align: left ;
1739     font-size: .85em ;
1740     margin-left: 1em ;
1741     border-top: 1px solid silver ;
1742 }
1743
1744 /* Remove footnote top border in the title page. */
1745 div.titlepage div.mpfootnotes {
1746     border-top: none ;
1747 }
1748
1749
```

```
1750
1751 ol {
1752   margin: 1ex 1em 1ex 0em;
1753   line-height: 1.2;
1754 }
1755
1756 ul, body dir, body menu {
1757   margin: 3ex 1em 3ex 0em;
1758   line-height: 1.2;
1759 }
1760
1761 li { margin: 0ex 0em 1ex 0em; }
1762
1763 html {
1764   margin: 0;
1765   padding: 0;
1766 }
1767
1768 .programlisting {
1769   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1770         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1771         "Courier New", monospace;
1772   margin: 1ex 0ex 1ex 0ex ;
1773   padding: .5ex 0pt .5ex 0pt ;
1774   overflow-x: auto;
1775 }
1776
1777 section.textbody>pre.programlisting {
1778 border-top: 1px solid silver ;
1779 border-bottom: 1px solid silver ;
1780 }
1781
1782
1783 div.displaymath {
1784     text-align: center ;
1785 }
1786
1787 div.displaymathnumbered {
1788     text-align: right ;
1789     margin-left: 5% ;
1790     margin-right: 5% ;
1791     min-width: 2.5in ;
1792 }
1793
1794 @media all and (min-width: 400px) {
1795     div.displaymathnumbered {
1796         margin-left: 10% ;
1797         margin-right: 10% ;
1798     }
1799 }
```

```
1800
1801 @media all and (min-width: 800px) {
1802     div.displaymathnumbered {
1803         margin-right: 20% ;
1804     }
1805 }
1806
1807 @media all and (min-width: 1200px) {
1808     div.displaymathnumbered {
1809         margin-right: 30% ;
1810     }
1811 }
1812
1813
1814 .inlineprogramlisting {
1815   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1816         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1817         "Courier New", monospace;
1818   overflow-x: auto;
1819 }
1820
1821 span.listinglabel {
1822     display: inline-block ;
1823     font-size: 70% ;
1824     width: 4em ;
1825     text-align: right ;
1826     margin-right: 2em ;
1827 }
1828
1829 div.abstract {
1830   margin: 2em 5% 2em 5% ;
1831   padding: 1ex 1em 1ex 1em ;
1832 /*  font-weight: bold ; */
1833   font-size: 90% ;
1834     text-align: left ;
1835 }
1836
1837 div.abstract dl {line-height:1.5;}
1838 div.abstract dt {color:#304070;}
1839
1840 div.abstracttitle{
1841     font-family: "URW Classico", Optima, "Linux Biolinum O",
1842         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1843         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1844     font-weight:bold;
1845     font-size:1.25em;
1846     text-align: center ;
1847 }
1848
1849 span.abstractrunintitle{
```

```
1850     font-family: "URW Classico", Optima, "Linux Biolinum O",
1851         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1852         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1853     font-weight:bold;
1854 }
1855
1856
1857 .verbatim {
1858     overflow-x: auto ;
1859 }
1860
1861 .alltt {
1862     overflow-x: auto ;
1863 }
1864
1865
1866 .bverbatim {
1867     margin: 1ex 0pt 1ex 0pt ;
1868     padding: .5ex 0pt .5ex 0pt ;
1869     overflow-x: auto ;
1870 }
1871
1872 .lverbatim {
1873     margin: 1ex 0pt 1ex 0pt ;
1874     padding: .5ex 0pt .5ex 0pt ;
1875     overflow-x: auto ;
1876 }
1877
1878 .fancyvrb {
1879     font-size:.85em ;
1880     margin: 3ex 0pt 3ex 0pt
1881 }
1882
1883 .fancyvrblabel {
1884     font-weight:bold;
1885     text-align: center ;
1886 }
1887
1888
1889 .verse {
1890     font-family: "Linux Libertine Mono O", "Lucida Console",
1891         "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1892         "Liberation Mono", "FreeMono", "Andale Mono",
1893         "Nimbus Mono L", "Courier New", monospace;
1894     margin-left: 1em ;
1895 }
1896
1897
1898 div.singlespace { line-height: 1.2 ; }
1899 div.onehalfspace { line-height: 1.5 ; }
```

```
div.doublespace { line-height: 2 ; }


/* Word processor format output: */
div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ;}




/* Minipage environments, vertically aligned to top, center, bottom: */
.minipage, .fminipage, .fcolorminipage {
    /* display: inline-block ; */
        /* Mini pages which follow each other will be tiled. */
    margin: .25em .25em .25em .25em;
    padding: .25em .25em .25em .25em;
    display: inline-flex;
    flex-direction: column ;
    overflow: auto;
}

/* Paragraphs in the flexbox did not collapse their margins. */
/* Have not yet researched this. */
.minipage p {margin: .75ex 0em .75ex 0em ;}

.fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
.fcolorbox .minipage, .fcolorboxBlock .minipage
    {border: none ; background: none;}

.fbox, .fboxBlock { border: 1px solid black ; }

.fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
.fminipage, .fcolorminipage
    {display: inline-block}

.shadowbox, .shabox {
   border: 1px solid black;
    box-shadow: 3px 3px 3px #808080 ;
     border-radius: 0px ;
    padding: .4ex .3em .4ex .3em ;
    margin: 0pt .3ex 0pt .3ex ;
  display: inline-block ;
}

.doublebox {
   border: 3px double black;
     border-radius: 0px ;
    padding: .4ex .3em .4ex .3em ;
    margin: 0pt .3ex 0pt .3ex ;
```

```
  display: inline-block ;
}

.ovalbox, .Ovalbox {
   border: 1px solid black;
     border-radius: 1ex ;
    padding: .4ex .3em .4ex .3em ;
    margin: 0pt .3ex 0pt .3ex ;
  display: inline-block ;
}

.Ovalbox { border-width: 2px ; }

.framebox {
   border: 1px solid black;
     border-radius: 0px ;
    padding: .3ex .2em 0ex .2em ;
    margin: 0pt .1ex 0pt .1ex ;
  display: inline-block ;
}


.mdframed {
/*     padding: 0ex ; */
/*    border: 1px solid blafck; */
/*      border-radius: 0px ; */
    padding: 0ex ;
    margin: 3ex 5% 3ex 5% ;
/*   display: inline-block ; */
}

.mdframed p { padding: 0ex .5em 0ex .5em ; }

.mdframed dl { padding: 0ex .5em 0ex .5em ; }

.mdframedtitle {
    padding: .5em ;
    display: block ;
    font-size: 130% ;
    margin-bottom: 1ex ;
}

.mdframedsubtitle {
    padding: 0ex .5em 0ex .5em ;
    display: block ;
    font-size: 115% ;
}

.mdframedsubsubtitle {
    padding: 0ex .5em 0ex .5em ;
```

```
2000     display: block ;
2001 }
2002
2003 .mdtheorem {
2004     padding: 0ex .5em 0ex .5em ;
2005     margin: 3ex 5% 3ex 5% ;
2006 /*   display: inline-block ; */
2007 }
2008
2009
2010 /* framed package */
2011 .framed, pre.boxedverbatim, fcolorbox {
2012     margin: 3ex 0em 3ex 0em ;
2013    border: 1px solid black;
2014      border-radius: 0px ;
2015     padding: .3ex 1em 0ex 1em ;
2016   display: block ;
2017 }
2018
2019 .shaded {
2020     margin: 3ex 0em 3ex 0em ;
2021     padding: .3ex 1em .3ex 1em ;
2022     display: block ;
2023 }
2024
2025 .snugframed {
2026     margin: 3ex 0em 3ex 0em ;
2027    border: 1px solid black;
2028      border-radius: 0px ;
2029   display: block ;
2030 }
2031
2032 .framedleftbar {
2033     margin: 3ex 0em 3ex 0em ;
2034    border-left: 3pt solid black;
2035      border-radius: 0px ;
2036     padding: .3ex .2em .3ex 1em ;
2037   display: block ;
2038 }
2039
2040 .framedtitle {
2041     margin: 0em ;
2042     padding: 0em ;
2043     font-size: 130%
2044 }
2045
2046 .framedtitle p { padding: .3em }
2047
2048
2049
```

```
dl {
  margin: 1ex 2em 1ex 0em;
  line-height: 1.3;
}

dl dt {
    margin-top: 1ex;
    margin-left: 1em ;
    font-weight: bold;
}

dl dd p { margin-top: 0em; }


nav {
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "DejaVu Sans", "Bitstream Vera Sans",
        Geneva, Verdana, sans-serif ;
    margin-bottom: 4ex ;
}

nav p {
    line-height: 1.2 ;
    margin-top:.5ex ;
    margin-bottom:.5ex;
    font-size: .9em ;
}



img, img.hyperimage, img.borderimage {
    max-width: 600px;
    border: 1px solid silver;
    box-shadow: 3px 3px 3px #808080 ;
    padding: .5% ;
    margin: .5% ;
    background: none ;
}

img.inlineimage{
    padding: 0px ;
    box-shadow: none ;
    border: none ;
    background: none ;
    margin: 0px ;
    display: inline-block ;
    border-radius: 0px ;
}

img.logoimage{
```

```
2100     max-width: 300px ;
2101     box-shadow: 3px 3px 3px #808080 ;
2102     border: 1px solid black ;
2103     background:none ;
2104     padding:0 ;
2105     margin:.5ex ;
2106     border-radius: 10px ;
2107 }
2108
2109
2110 .section {
2111 /*
2112     To have each section float relative to each other:
2113 */
2114 /*
2115     display: block ;
2116     float: left ;
2117     position: relative ;
2118     background: white ;
2119     border: 1px solid silver ;
2120     padding: .5em ;
2121 */
2122     margin: 0ex .5em 0ex .5em ;
2123     padding: 0 ;
2124 }
2125
2126
2127 figure {
2128     margin: 5ex auto 5ex auto ;
2129     padding: 1ex 1em 1ex 1em ;
2130     overflow-x: auto ;
2131 }
2132
2133
2134 /* To automatically center images in figures: */
2135 /*
2136 figure img.inlineimage {
2137     margin: 0ex auto 0ex auto ;
2138     display: block ;
2139 }
2140 */
2141
2142 /* To automatically center minipages in figures: */
2143 /*
2144 figure div.minipage, figure div.minipage div.minipage {
2145     margin: 1ex auto 1ex auto ;
2146     display: block ;
2147 }
2148 */
2149
```

```
figure div.minipage p { font-size: 85% ; }

figure.subfigure, figure.subtable  {
    display: inline-block ; margin: 3ex 1em 3ex 1em ;
}

figcaption .minipage { margin:0 ; padding: 0 }

div.minipage figure { border: none ; box-shadow: none ; }

div.floatrow { text-align: center; }

div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }

div.floatfoot { font-size: .85em ;
    border-top: 1px solid silver ; line-height: 1.2 ; }

figcaption , .lstlistingtitle {
    font-size: .85em ;
    text-align: center ;
    font-weight: bold ;
    margin-top: 1ex ;
    margin-bottom: 1ex ;
}

figure.subfigure figcaption, figure.subtable figcaption  {
    border-bottom: none ; background: none ;
}

div.nonfloatcaption {
    margin: 1ex auto 1ex auto ;
    font-size: .85em ;
    text-align: center ;
    font-weight: bold ;
}

/* For a \RawCaption inside a minipage inside a figure's floatrow: */
figure div.floatrow div.minipage figcaption {
    border: none ;
    background: none ;
}


/* For packages such as float, rotfloat, and algorithm2e: */

figure.boxed, figure.boxruled {
    border: 1px solid black ;
}

figure.ruled {
```

```
    border-top: 1px solid black ;
    border-bottom: 1px solid black ;
    border-left: 0px ;
    border-right: 0px ;
    border-radius: 0px ;
    background: none ;
    box-shadow: none ;
}

figure.ruled figcaption, figure.boxruled figcaption {
    border-top: 1px solid silver ;
    border-bottom: 1px solid silver ;
}


table {
    margin: 1ex auto 1ex auto ;
    border-collapse: separate ;
    border-spacing: 0px ;
    line-height: 1.3 ;
    }

tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
    margin-bottom: 0ex ; } /* for \hline */

tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
    margin-bottom: 0ex ; } /* for \toprule, \bottomrule */

td {padding: .5ex .5em .5ex .5em ;}

table td.tdl { text-align: left ; vertical-align: middle ; }
table td.tdc { text-align: center ; vertical-align: middle ; }
table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
table td.tdbang { text-align: center ; vertical-align: middle ; }
table td.tdr { text-align: right ; vertical-align: middle ; }
table td.tdp { text-align: left ; vertical-align: bottom ; }
table td.tdm { text-align: left ; vertical-align: middle ; }
table td.tdb { text-align: left ; vertical-align: top ; }
table td.tdP { text-align: center ; vertical-align: bottom ; }
table td.tdM { text-align: center ; vertical-align: middle ; }
table td.tdB { text-align: center ; vertical-align: top ; }

table td.tvertbarl { border-left: 1px solid black }
table td.tvertbarldouble { border-left: 4px double black }
table td.tvertbarr { border-right: 1px solid black }
table td.tvertbarrdouble { border-right: 4px double black }

table td.tvertbarldash { border-left: 1px dashed black }
table td.tvertbarldoubledash { border-left: 2px dashed black }
table td.tvertbarrdash { border-right: 1px dashed black }
```

```
table td.tvertbarrdoubledash { border-right: 2px dashed black }


/* for cmidrules: */
table td.tdrule {
    border-top: 1px solid #A0A0A0 ;
}

table td.tdrulel {
    border-top-left-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}

table td.tdruler {
    border-top-right-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}

table td.tdrulelr {
    border-top-left-radius:.5em ;
    border-top-right-radius:.5em ;
    border-top: 1px solid #A0A0A0 ;
}


/* Margins of paragraphs inside table cells: */
td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }
td.tdm p , td.tmbrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }
td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
    padding-bottom: 1ex ; margin: 0ex ; }

td.tdp , td.tdprule , td.tdP , td.tdPrule
    { padding: 0ex .5em 0ex .5em ; }
td.tdm , td.tdmrule , td.tdM , td.tdMrule
    { padding: 0ex .5em 0ex .5em ; }
td.tdb , td.tdbrule , td.tdB , td.tdBrule
    { padding: 0ex .5em 0ex .5em ; }


/* table notes: */
.tnotes {
    margin: 0ex 5% 1ex 5% ;
    padding: 0.5ex 1em 0.5ex 1em;
    font-size:.85em;
    text-align: left ;
}

.tnotes dl dt p {margin-bottom:0px;}
```

```
2300
2301 .tnoteitemheader {margin-right: 1em;}
2302
2303
2304 /* for colortbl and cell color */
2305 div.cellcolor {
2306     width: 100% ;
2307     padding: .5ex .5em .5ex .5em ;
2308     margin: -.5ex -.5em -.5ex -.5em ;
2309 }
2310
2311
2312 /* for bigdelim */
2313 .ldelim, .rdelim { font-size: 200% }
2314
2315
2316 /* center, flushleft, flushright environments */
2317 div.center{text-align:center;}
2318 div.center table {margin-left:auto;margin-right:auto;}
2319 div.flushleft{text-align:left;}
2320 div.flushleft table {margin-left:0em ; margin-right:auto;}
2321 div.flushright{text-align:right;}
2322 div.flushright table {margin-left:auto ; margin-right: 0em ;}
2323
2324
2325 /* Fancybox */
2326 div.Btrivlist table tr td {
2327     padding: .2ex 0em ;
2328 }
2329
2330
2331 /* program listing callouts: */
2332 span.callout {
2333      font-family: "DejaVu Sans", "Bitstream Vera Sans",
2334         Geneva, Verdana, sans-serif ;
2335     border-radius: .5em;
2336     background-color:black;
2337     color:white;
2338     padding:0px .25em 0px .25em;
2339     margin: 0 ;
2340     font-weight: bold;
2341     font-size:.72em ;
2342 }
2343
2344 div.programlisting pre.verbatim span.callout{
2345     font-size: .85em ;
2346 }
2347
2348 span.verbatim {
2349     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
```

```
2350         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
2351         "Courier New", monospace;
2352 }
2353
2354
2355
2356 div.published
2357 {
2358     text-align: center ;
2359     font-variant: normal ;
2360     font-style: italic ;
2361     font-size: 1em ;
2362     margin: 3ex 0em 3ex 0em ;
2363 }
2364
2365 div.subtitle
2366 {
2367     text-align: center ;
2368     font-variant: normal ;
2369     font-style: italic ;
2370     font-size: 1.25em ;
2371     margin: 3ex 0em 3ex 0em ;
2372 }
2373
2374 div.subtitle p { margin: 1ex ; }
2375
2376 div.author
2377 {
2378     font-variant: normal ;
2379     font-style: normal ;
2380     font-size: 1em ;
2381     margin: 3ex 0em 3ex 0em ;
2382 }
2383
2384 div.oneauthor {
2385     display: inline-block ;
2386     margin: 3ex 1em 0ex 1em ;
2387 }
2388
2389 /*
2390 div.author table {
2391     margin: 3ex auto 0ex auto ;
2392     background: none ;
2393 }
2394
2395 div.author table tbody tr td { padding: .25ex ; }
2396 */
2397
2398 span.affiliation {font-size: .85em ; font-variant: small-caps; }
2399
```

```
2400 div.titledate {
2401     text-align: center ;
2402     font-size: .85em ;
2403     font-style: italic;
2404     margin: 6ex 0em 6ex 0em ;
2405 }
2406
2407
2408 nav.topnavigation{
2409     text-align: left ;
2410     padding: 0.5ex 1em 0.5ex 1em ;
2411 /*     margin: 2ex 0em 3ex 0em ; */
2412     margin: 0 ;
2413     border-bottom: 1px solid silver ;
2414     border-top: 1px solid silver ;
2415     clear:right ;
2416 }
2417
2418 nav.botnavigation{
2419     text-align: left ;
2420     padding: 0.5ex 1em 0.5ex 1em ;
2421 /*     margin: 3ex 0em 2ex 0em ; */
2422     margin: 0 ;
2423     border-top: 1px solid silver ;
2424     border-bottom: 1px solid silver ;
2425     clear:right ;
2426 }
2427
2428
2429 header{
2430     line-height: 1.2 ;
2431     font-size: 1em ;
2432 /*     border-bottom: 2px solid silver ; */
2433     margin: 0px ;
2434     padding: 0ex 1em 0ex 1em ;
2435     text-align:center ;
2436 }
2437
2438 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
2439
2440
2441 footer{
2442     font-size: .85em ;
2443     line-height: 1.2 ;
2444     margin-top: 1ex ;
2445     border-top: 2px solid silver ;
2446     padding: 2ex 1em 2ex 1em ;
2447     clear:right ;
2448     text-align:left ;
2449 }
```

```
2450
2451
2452 a.linkhome { font-weight:bold ; font-size: 1em ;}
2453
2454
2455 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
2456
2457 img.lateximage{
2458     padding: 0pt ;
2459     margin: 0pt ;
2460     box-shadow: none ;
2461     border: none ;
2462     background: none ;
2463     max-width: 100% ;
2464     border-radius: 0ex ;
2465     border: none ;
2466 }
2467
2468
2469
2470 nav.sidetoc {
2471     font-family: "DejaVu Serif", "Bitstream Vera Serif",
2472         "Lucida Bright", Georgia, serif;
2473     float:right ;
2474     width: 20%;
2475     border-left: 1px solid silver;
2476     border-top: 1px solid silver;
2477     border-bottom: 1px solid silver;
2478 /*     border-top: 2px solid #808080 ; */
2479     background: #FAF7F4 ;
2480     padding: 2ex 0em 2ex 1em ;
2481     margin: 0ex 0em 2ex 1em ;
2482     font-size:.9em ;
2483     border-radius: 20px 0px 0px 20px ;
2484     }
2485
2486 div.sidetoccontents {
2487 /*     border-top: 1px solid silver ; */
2488     overflow-y: auto ;
2489     width: 100% ;
2490     text-align: left ;
2491 }
2492
2493
2494 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
2495     text-indent: 0 ; }
2496
2497 nav.sidetoc p a {color:black ; font-size: .7em ;}
2498
2499 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
```

```
    border-bottom: 1px solid silver ;    }

nav.sidetoc a:hover {text-decoration: underline ; }



section.textbody { margin: 0ex 1em 0ex 1em ;}


div.multicolsheading { -webkit-column-span: all;
    -moz-column-span: all; column-span: all; }
div.multicols { -webkit-columns: 3 380px ;
    -moz-columns: 3 380px ; columns: 3 380px ; }
div.multicols p {margin-top: 0ex}


/* Used for algorithm2e: */
div.alg2evline{
    margin-left: 1em ;
    padding-left: 1em ;
    border-left: 1px solid black ;
    border-radius: 0px 0px 0px 1ex ;
}

div.alg2evsline{
    margin-left: 1em ;
    padding-left: 1em ;
    border-left: 1px solid black ;
}

div.alg2enoline{
    margin-left: 1em ;
    padding-left: 1em ;
}

span.alg2elinenumber{
    margin-right: .5em ;
    font-size: 50% ;
    color: red ;
}


/* Used for algorithmicx: */
span.floatright { float: right ; }



/* Native LaTeX theorems: */

```

```
.theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
.theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }


/* theorem, amsthm, and ntheorem packages */

span.theoremheader,
span.theoremheaderplain,
span.theoremheaderdefinition,
span.theoremheaderbreak,
span.theoremheadermarginbreak,
span.theoremheaderchangebreak,
span.theoremheaderchange,
span.theoremheadermargin
{
    font-style:normal ; font-weight: bold ; margin-right: 1em ;
}

span.amsthmnameplain,
span.amsthmnamedefinition,
span.amsthmnumberplain,
span.amsthmnumberdefinition
{
    font-style:normal ; font-weight: bold ;
}


span.amsthmnameremark,
span.amsthmnumberremark
{font-style:italic ; font-weight: normal ; }


span.amsthmnoteplain,
span.amsthmnotedefinition
{font-style:normal ;}


span.theoremheaderremark,
span.theoremheaderproof,
span.amsthmproofname
{font-style:italic ; font-weight: normal ; margin-right: 1em ; }

span.theoremheadersc
{
    font-style:normal ;
    font-variant: small-caps ;
    font-weight: normal ;
    margin-right: 1em ;
}

```

```
.theoremendmark {float:right}

div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
div.theorembodybreak, div.theorembodynonumberbreak,
div.theorembodymarginbreak,
div.theorembodychangebreak,
div.theorembodychange,
div.theorembodymargin
{
    font-style:italic;
    margin-top: 3ex ; margin-bottom: 3ex ;
}

div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
div.theorembodyplainupright, nonumberplainuprightsc,
div.amsthmbodydefinition, div.amsthmbodyremark,
div.amsthmproof
{
    font-style: normal ;
    margin-top: 3ex ; margin-bottom: 3ex ;
}

span.amsthmnoteremark {}



/*
For CSS LaTeX and related logos:
Based on:
http://edward.oconnor.cx/2007/08/tex-poshlet
http://nitens.org/taraborelli/texlogo
*/

.latexlogofont {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    font-variant: normal ;
}

.latexlogo {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    letter-spacing: .03em ;
    font-size: 1.1em;
}

.latexlogo sup {
  text-transform: uppercase;
  letter-spacing: .03em ;
  font-size: 0.85em;
```

```
  vertical-align: 0.15em;
  margin-left: -0.36em;
  margin-right: -0.15em;
}

.latexlogo sub {
  text-transform: uppercase;
  vertical-align: -0.5ex;
  margin-left: -0.1667em;
  margin-right: -0.125em;
  font-size: 1em;
}

.xetexlogo {
    font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
        "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
    letter-spacing: .03em ;
    font-size: 1.1em;
}

/* A smaller gap between Xe and Tex v.s. LaTeX: */
.xetexlogo sub {
  text-transform: uppercase;
  vertical-align: -0.5ex;
  margin-left: -0.0667em;
  margin-right: -0.2em;
  font-size: 1em;
  letter-spacing: .03em ;
}

/* A large gap between Xe and LaTeX v.s. TeX: */
.xelatexlogo sub {
  text-transform: uppercase;
  vertical-align: -0.5ex;
  margin-left: -0.0667em;
  margin-right: -.05em;
  font-size: 1em;
  letter-spacing: .03em ;
}

.amslogo {
    font-family: "TeXGyreChorus","URW Chancery L",
        "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
        "Linux Libertine O", "Nimbus Roman No 9 L",  "FreeSerif",
        "Hoefler Text", Times, "Times New Roman", serif;
  font-style: italic;
}

.lyxlogo {
    font-family: "URW Classico", Optima, "Linux Biolinum O",
```

```
2700         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2701         Verdana, sans-serif ;
2702 }
2703
2704
2705 /* Only display top and bottom navigation if a small screen: */
2706 /* Hide the sidetoc if a small screen: */
2707 nav.topnavigation { display:none; }
2708 nav.botnavigation { display:none; }
2709
2710 @media screen and (max-width: 45em) {
2711 /*     nav.sidetoc {display:none;} */
2712     nav.sidetoc {
2713         float: none ;
2714         width: 100% ;
2715         margin: 5ex 0px 5ex 0px ;
2716         padding: 0 ;
2717         border-radius: 0 ;
2718         border-bottom: 1px solid black ;
2719         border-top: 1px solid black ;
2720         box-shadow: none ;
2721     }
2722 /*     nav.topnavigation { display:block } */
2723     nav.botnavigation { display:block }
2724     .marginpar {
2725         max-width: 100%;
2726         float: none;
2727         display:block ;
2728         margin: 1ex 1em 1ex 1em ;
2729     }
2730 }
2731
2732 @media print {
2733     body {
2734         font-family: "Linux Libertine O",
2735         "DejaVu Serif", "Bitstream Vera Serif",
2736         "Liberation Serif", "Nimbus Roman No 9 L",
2737         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2738     }
2739     nav.sidetoc { display:none; }
2740     nav.topnavigation { display: none; }
2741     nav.botnavigation { display: none; }
2742 }
2743
2744 @media handheld {
2745     nav.sidetoc { display:none; }
2746     nav.topnavigation { display:block }
2747     nav.botnavigation { display:block }
2748 }
2749
```

```
@media projection {
    nav.sidetoc { display:none; }
    nav.topnavigation { display:block }
    nav.botnavigation { display:block }
}
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

37.4 lwarp_sagebrush.css

File `lwarp_sagebrush.css` An optional CSS which may be used for a semi-modern appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_sagebrush.css}
@import url("lwarp.css") ;


A:link {color:#105030 ; text-decoration: none ; }
A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}



h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
{
    font-family: "URW Classico", Optima, "Linux Biolinum O",
        "Linux Libertine O", "Liberation Serif",
        "Nimbus Roman No 9 L",  "FreeSerif",
        "Hoefler Text", Times, "Times New Roman", serif;
    font-variant: small-caps ;
    font-weight: normal ;
    color: #304070 ;
    text-shadow: 2px 2px 3px #808080;
}

h1 {    /* title of the entire website, used on each page */
    font-variant: small-caps ;
    color: #304070 ;
    text-shadow: 2px 2px 3px #808080;
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
}
```

```
2789
2790 h1 {
2791   border-bottom: 1px solid #304070;
2792   border-top: 2px solid #304070;
2793 }
2794
2795 h2 {
2796   border-bottom: 1px solid #304070;
2797   border-top: 2px solid #304070;
2798     background-color: #F7F7F0 ;
2799     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2800 }
2801
2802
2803
2804 div.abstract {
2805     background: #f5f5eb ;
2806     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2807
2808   border: 1px solid silver;
2809     border-radius: 1em ;
2810 }
2811
2812 div.abstract dl {line-height:1.5;}
2813 div.abstract dt {color:#304070;}
2814
2815 div.abstracttitle{
2816     font-family: "URW Classico", Optima, "Linux Biolinum O",
2817         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2818         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2819     font-weight:bold;
2820     font-variant: small-caps ;
2821     font-size:1.5em;
2822     border-bottom: 1px solid silver ;
2823     color: #304070 ;
2824     text-align: center ;
2825     text-shadow: 1px 1px 2px #808080;
2826 }
2827
2828 span.abstractrunintitle{
2829     font-family: "URW Classico", Optima, "Linux Biolinum O",
2830         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2831         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2832     font-weight:bold;
2833 }
2834
2835
2836 div.epigraph, div.dictum {
2837     background: #f5f5eb ;
2838     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
```

```
2839
2840     border: 1px solid silver ;
2841     border-radius: 1ex ;
2842     box-shadow: 3px 3px 3px #808080 ;
2843 }
2844
2845
2846 .example {
2847     background-color: #f5f5eb ;
2848     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2849
2850 }
2851
2852 div.exampletitle{
2853     font-family: "URW Classico", Optima, "Linux Biolinum O",
2854         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2855         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2856     font-weight:bold;
2857     font-variant: small-caps ;
2858     border-bottom: 1px solid silver ;
2859     color: #304070 ;
2860     text-align: center ;
2861     text-shadow: 1px 1px 2px #808080;
2862 }
2863
2864
2865 .sidebar {
2866     background-color: #f5f5eb ;
2867     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2868
2869 }
2870
2871 div.sidebartitle{
2872     font-family: "URW Classico", Optima, "Linux Biolinum O",
2873         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2874         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2875     font-weight:bold;
2876     font-variant: small-caps ;
2877     border-bottom: 1px solid silver ;
2878     color: #304070 ;
2879     text-align: center ;
2880     text-shadow: 1px 1px 2px #808080;
2881 }
2882
2883
2884 .fancyvrblabel {
2885     font-family: "URW Classico", Optima, "Linux Biolinum O",
2886         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2887         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2888     font-weight:bold;
```

```
2889     font-variant: small-caps ;
2890     font-size: 1.5em ;
2891     color: #304070 ;
2892     text-align: center ;
2893     text-shadow: 1px 1px 2px #808080;
2894 }
2895
2896 div.minipage {
2897     background-color: #eeeee7 ;
2898     border: 1px solid silver ;
2899     border-radius: 1ex ;
2900 }
2901
2902 table div.minipage { background: none ; border: none ; }
2903
2904 div.framebox div.minipage {border:none ; background:none}
2905
2906 section.textbody > div.minipage {
2907     box-shadow: 3px 3px 3px #808080 ;
2908 }
2909
2910 div.fboxBlock div.minipage { box-shadow: none ; }
2911
2912 .framed .minipage , .framedleftbar .minipage {
2913     border: none ;
2914     background: none ;
2915     padding: 0ex ;
2916     margin: 0ex ;
2917 }
2918
2919 figure.figure .minipage, figcaption .minipage { border: none; }
2920
2921 div.marginblock div.minipage ,
2922 div.marginparblock div.minipage
2923     { border: none; }
2924
2925 figure , div.marginblock {
2926     background-color: #eeeee7 ;
2927     border: 1px solid silver ;
2928     border-radius: 1ex ;
2929     box-shadow: 3px 3px 3px #808080 ;
2930 }
2931
2932 figure figure {
2933     border: 1px solid silver ;
2934     margin: 0em ;
2935     box-shadow: none ;
2936 }
2937
2938 /*
```

```
figcaption {
    border-top: 1px solid silver ;
    border-bottom: 1px solid silver ;
    background-color: #e8e8e8 ;
}
*/


div.table {
    box-shadow: 3px 3px 3px #808080 ;
}

/*
.tnotes {
    background: #e8e8e8;
    border: 1px solid silver;
}
*/


nav.topnavigation{
    background-color: #b0b8b0 ;
    background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
}

nav.botnavigation{
    background-color: #b0b8b0 ;
    background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
}



header{
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
}

footer{
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
}



nav.sidetoc {
    background-color: #F7F7F0 ;
    background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
    box-shadow: 3px 3px 3px #808080 ;
    border-radius: 0px 0px 0px 20px ;
    }
```

```
2989
2990 div.sidetoctitle {color: #304070 ; }
2991
2992 nav.sidetoc a:hover {
2993     color:#006000 ;
2994     text-decoration: none ;
2995     text-shadow:0px 0px 2px #a0a0a0;
2996 }
2997
2998
2999 @media screen and (max-width: 45em) {
3000     nav.sidetoc { border-radius: 0 ; }
3001 }
3002
3003
3004 \end{filecontents*}
3005 % \end{Verbatim}% for syntax highlighting
3006 \end{warpprint}
```

37.5 lwarp_formal.css

File `lwarp_formal.css` An optional CSS which may be used for a more formal appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
3007 \begin{warpprint}
3008 \begin{filecontents*}{lwarp_formal.css}
3009 @import url("lwarp.css") ;
3010
3011
3012
3013 A:link {color:#802020 ; text-decoration:none; }
3014 A:visited {color:#802020 ; text-shadow:none ;}
3015 A:hover {color:#400000 ; text-shadow:none ;}
3016 A:active {color:#C00000 ; text-shadow:none ;}
3017
3018
3019 body {
3020     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3021         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3022         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
3023         "Times New Roman", serif;
3024     background: #fffcf5;
3025 }
3026
3027 span.textrm {
```

```
3028     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3029         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3030         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
3031         "Times New Roman", serif;
3032 }
3033
3034 span.textsf {
3035      font-family: "DejaVu Sans", "Bitstream Vera Sans",
3036         Geneva, Verdana, sans-serif ;
3037 }
3038
3039
3040
3041 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
3042 {
3043     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3044         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3045         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
3046         "Times New Roman", serif;
3047     color: #800000 ;
3048     text-shadow: none ;
3049 }
3050
3051 h1, h2 {
3052     background-color: #fffcf5 ;
3053     background-image: none ;
3054     border-bottom: 1px solid #808080;
3055     border-top: 2px solid #808080;
3056 }
3057
3058 div.abstracttitle {
3059     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3060         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3061         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
3062         "Times New Roman", serif;
3063     color: black ;
3064     text-shadow: none ;
3065 }
3066
3067 span.abstractrunintitle {
3068     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3069         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3070         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
3071         "Times New Roman", serif;
3072     color: black ;
3073     text-shadow: none ;
3074 }
3075
3076 div.abstract { font-size: 100% }
3077
```

```
.sidebar {
    background: #fffcf5;
    background-image: none ;
  margin: 2em 5% 2em 5%;
  padding: 0.5em 1em;
  border: none ;
  border-top : 1px solid silver;
  border-bottom : 1px solid silver;
  font-size: 90% ;
}

div.sidebartitle{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: #800000 ;
    text-shadow: none ;
    border: none ;
}

.example {
    background: #fffcf5;
    background-image: none ;
  margin: 2em 5% 2em 5%;
  padding: 0.5em 1em;
  border: none ;
  border-top : 1px solid silver;
  border-bottom : 1px solid silver;
}

div.exampletitle{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: #800000 ;
    text-shadow: none ;
    border: none ;
}

div.fancyvrblabel{
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    color: #800000 ;
    text-shadow: none ;
    border: none ;
}
```

```
3128
3129
3130
3131 .verse {
3132     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3133         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
3134         "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
3135         "Times New Roman", serif;
3136 }
3137
3138
3139 figure {
3140     margin: 5ex 5% 5ex 5% ;
3141     padding: 1ex 1em 1ex 1em ;
3142     background-color: #fffcf5 ;
3143     overflow-x: auto ;
3144     border: none ;
3145 /*     border-top: 1px solid silver; */
3146 /*     border-bottom: 1px solid silver; */
3147 }
3148
3149
3150 figcaption , .lstlisting {
3151     border: none ;
3152 /*     border-top: 1px solid silver ; */
3153 /*     border-bottom: 1px solid silver ; */
3154     background-color: #fffcf5 ;
3155 }
3156
3157 .tnotes {
3158     background: #fffcf5 ;
3159 }
3160
3161 .theorem {
3162         background: none ;
3163 }
3164
3165 .minipage {
3166     background-color: #fffcf5 ;
3167     border: none ;
3168 }
3169
3170 div.floatrow figure { border: none ; }
3171
3172 figure figure { border: none ; }
3173
3174
3175 nav.toc, nav.lof, nav.lot, nav.lol {
3176     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
3177         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
```

```
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
}

nav.sidetoc {
    font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
        "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
        "Liberation Serif", "Nimbus Roman No 9 L",  "FreeSerif", Times,
        "Times New Roman", serif;
    background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
    border-radius: 0px 0px 0px 20px ;
}

div.sidetoctitle{
    color: #800000 ;
}

header{
    background-color: #e0e0e0 ;
    background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
    text-align:center ;
}

footer{
    background-color: #e0e0e0 ;
    background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
    padding: 2ex 1em 2ex 1em ;
    clear:right ;
    text-align:left ;
}

nav.botnavigation {
    background: #dedcd5 ;
    border-top: 1px solid black ;
}
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

37.6 sample_project.css

File `sample_project.css` The project-specific css file. Use with `\CSSFilename`.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
\begin{warpprint}
```

```
\begin{filecontents*}{sample_project.css}
/* ( --- Start of project.css --- ) */
/* ( --- A sample project-specific CSS file for lwarp --- ) */

/* Uncomment one of the following: */
@import url("lwarp.css") ;
/* @import url("lwarp_formal.css") ; */
/* @import url("lwarp_sagebrush.css") ; */

/* Project-specific CSS setting follow here. */
/* . . . */

/* ( --- End of project.css --- ) */
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

37.7 lwarp.ist

File `lwarp.ist` Used to modify the index for **lwarp**.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

The page compositor line is for **memoir**'s `\specialindex`.

```
\begin{warpprint}
\begin{filecontents*}{lwarp.ist}
preamble
"\\begin{theindex}
  \\providecommand*\\lettergroupDefault[1]{}
  \\providecommand*\\lettergroup[1]{%
      \\par\\textbf{#1}\\par
      \\nopagebreak
  }
"
headings_flag 1
heading_prefix "
  \\lettergroup{"
heading_suffix "}"
delim_0 ", \\hyperindexref{"
delim_1 ", \\hyperindexref{"
delim_2 ", \\hyperindexref{"
delim_n "}, \\hyperindexref{"
delim_r "} -- \\hyperindexref{"
delim_t "}"
page_compositor "."
```

```
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

37.8 lwarp.xdy

File `lwarp.xdy` Used to modify the index for **lwarp**.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

See:
https://tex.stackexchange.com/questions/80300/how-can-i-convince-hyperref-and-xindy-to-play-together-nicely

```
\begin{warpprint}
\begin{filecontents*}{lwarp.xdy}
(require "tex/inputenc/latin.xdy")
(merge-rule "\\PS *" "Postscript")
(require "texindy.xdy")
(require "page-ranges.xdy")
(require "book-order.xdy")
(define-location-class "arabic-page-numbers"
    ("arabic-numbers") :min-range-length 1)
(require "makeindex.xdy")
(define-attributes (("hyperindexref")))
(markup-locref :open "\hyperindexref{" :close "}")
(markup-locref :open "\hyperindexref{" :close "}" :attr "hyperpage")
(markup-locref :open "\textbf{\hyperindexref{" :close "}}" :attr "textbf")
(markup-locref :open "\textit{\hyperindexref{" :close "}}" :attr "textit")
(define-location-class-order ("roman-page-numbers"
                  "arabic-page-numbers"
                  "alpha-page-numbers"
                  "Roman-page-numbers"
                  "Alpha-page-numbers"
                  "see"
                  "seealso"))
\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

37.9 lwarp_one_limage.cmd

File `lwarp_one_limage.cmd` Used by **lwarp** to help make `lateximages` when using WINDOWS.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

The arguments are each of the three fields from `lateximages.txt`, and also the base name of the source file.

MikTeX does not allow file `lwarp_one_limage.cmd` to be created directly by **lwarpmk**, so `lwarp_one_limage.txt` is created instead, then copied to `lwarp_one_limage.cmd` by **lwarpmk**. This occurs each time **lwarpmk** used to create `lateximages`.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_one_limage.txt}
@echo off
pdfseparate -f %1 -l %1 %4_html.pdf lateximages\lateximagetemp-%%d.pdf
pdfcrop  --hires  lateximages\lateximagetemp-%1.pdf lateximages\%3.pdf
pdftocairo  -svg -noshrink  lateximages\%3.pdf lateximages\%3.svg
del lateximages\%3.pdf
del lateximages\lateximagetemp-%1.pdf
exit
\end{filecontents*}
\end{warpprint}
```

37.10 lwarp_mathjax.txt

File `lwarp_mathjax.txt` Used by **lwarp** when using MATHJAX.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```
\begin{warpprint}
\begin{filecontents*}{lwarp_mathjax.txt}
<!-- https://groups.google.com/forum/#!topic/
                                mathjax-users/jUtewUcE2bY -->
<script type="text/x-mathjax-config">
MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
    var seteqsectionDefault = {name: "", num: 0};
    var seteqsections = {}, seteqsection = seteqsectionDefault;
    var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
    var AMS = MathJax.Extension["TeX/AMSmath"];
    TEX.Definitions.Add({
    macros: {
        seteqsection: "mySection",
        seteqnumber: "mySetEqNumber"
    }
    });

    PARSE.Augment({
```

```
    mySection: function (name) {
        seteqsection.num = AMS.number;
        var n = this.GetArgument(name);
        if (n === "") {
            seteqsection = seteqsectionDefault;
        } else {
            if (!seteqsections["_"+n])
                seteqsections["_"+n] = {name:n, num:0};
            seteqsection = seteqsections["_"+n];
        }
        AMS.number = seteqsection.num;
    },
    mySetEqNumber: function (name) {
        var n = this.GetArgument(name);
        if (!n || !n.match(/^ *[0-9]+ *$/))
            n = "";
        else
            n = parseInt(n)-1;
        <!-- $ syntax highlighting -->
        if (n === "" || n < 1)
            TEX.Error
            ("Argument to "+name+" should be a positive integer");
        AMS.number = n;
    }
    });
    MathJax.Hub.Config({
    TeX: {
        equationNumbers: {
            formatTag: function (n) {
                <!-- if not numeric, don't include the chapter -->
                if (!n.match(/^ *[0-9]+ *$/ ))
                <!-- $ syntax highlighting -->
                    return "("+(n).replace(/^\./,"")+")" ;
                else
                    return "("+(seteqsection.name+"."+n).replace(/^\./,"")+")" ;
            },
            formatID: function (n) {
                n = (seteqsection.name+'.'+n).replace
                    (/[:"'<>&]/g,"").replace(/^\./,"");
                return 'mjx-eqn-' + n;
            }
        }
    }
    });
});
</script>

<!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
<script type="text/x-mathjax-config">
  MathJax.Ajax.config.path["Contrib"] =
```

```
    "https://cdn.mathjax.org/mathjax/contrib";
</script>

<!-- https://github.com/burnpanck/MathJax-siunitx -->

<script type="text/x-mathjax-config">
 MathJax.Hub.Config({
   extensions: ["tex2jax.js","[siunitx]/siunitx.js"],
   jax: ["input/TeX","output/HTML-CSS"],
   tex2jax: {
        inlineMath: [["$","$"],["\\(","\\)"]] ,
        processClass: "tabbing|verse"
    },
   TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "sinuitx.js"]}
 });
 MathJax.Ajax.config.path['siunitx']  = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
 </script>

<script type="text/x-mathjax-config">
MathJax.Hub.Config({
    TeX: {
    equationNumbers: {
        autoNumber: "AMS"
    }
    }
});
</script>

<!-- Alternative CDN provider: -->
<script type="text/javascript" async
src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.4/MathJax.js?config=TeX-AMS_HTML-full">
</script>

<!-- No longer supported after April 30, 2017: -->
<!--
<script
  src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
</script>
-->

\end{filecontents*}
% \end{Verbatim}% for syntax highlighting
\end{warpprint}
```

37.11 lwarpmk.lua — lwarpmk option

Opt `lwarpmk` Creates a local copy of **lwarpmk**.

Prog `lwarpmk` Command-line utility to process **lwarp** files and images.

parallel processing `lateximages` and SVG math images are generated using multiple processes in parallel. For UNIX and LINUX, every 32 images the `wait` command is issued to wait for the previous batch of images to finish processing before starting a new batch. For WINDOWS, every 32 images one task is dispatched with

```
START /B /WAIT /BELOWNORMAL
```

which causes the operating system to wait until this lesser-priority tasks finishes, hopefully also waiting for the normal priority tasks which were already in progress to also complete. Afterwards, the next batch of images is started.

The following is only generated if the `lwarpmk` option was given to **lwarp**.

```
\begin{LWR@createlwarpmk}

\begin{filecontents*}{lwarpmk.lua}
#!/usr/bin/env texlua

-- Copyright 2016-2018 Brian Dunn


printversion = "v0.60"
requiredconfversion = "1" -- also at *lwarpmk.conf

function printhelp ()
print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
end


function printusage ()
--
-- Print the usage of the lwarpmk command:
--
print ( [[

lwarpmk print [-p project]: Compile the print version if necessary.
lwarpmk print1 [-p project]: Forced single compile of the print version.
lwarpmk printindex [-p project]: Process print indexes.
lwarpmk printglossary [-p project]: Process the glossary for the print version.
lwarpmk html [-p project]: Compile the HTML version if necessary.
lwarpmk html1 [-p project]: Forced single compile of the HTML version.
lwarpmk htmlindex [-p project]: Process HTML indexes.
lwarpmk htmlglossary [-p project]: Process the glossary for the html version.
lwarpmk again [-p project]: Touch the source code to trigger recompiles.
lwarpmk limages [-p project]: Process the "lateximages" created by lwarp.sty.
lwarpmk pdftohtml [-p project]:
    For use with latexmk or a Makefile:
```

```
3437     Converts project_html.pdf to project_html.html and individual HTML files.
3438     Finishes the HTML conversion even if there was a compile error.
3439 lwarpmk pdftosvg <list of file names>: Converts each PDF file to SVG.
3440 lwarpmk epstopdf <list of file names>: Converts each EPS file to PDF.
3441 lwarpmk clean [-p project]: Remove .aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
3442 lwarpmk cleanall [-p project]: Remove auxiliary files and also project.pdf, *.html
3443 lwarpmk cleanlimages: Removes all images from the "lateximages" directory.
3444 lwarpmk -h: Print this help message.
3445 lwarpmk --help: Print this help message.
3446
3447 ]] )
3448 -- printconf ()
3449 end
3450
3451
3452 function splitfile (destfile,sourcefile)
3453 --
3454 -- Split one large sourcefile into a number of files,
3455 -- starting with destfile.
3456 -- The file is split at each occurance of <!--|Start file|newfilename|*
3457 --
3458 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
3459 local sfile = io.open(sourcefile)
3460 io.output(destfile)
3461 for line in sfile:lines() do
3462 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
3463 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then
3464     -- split the file
3465     io.output(newfilename) ;
3466 else
3467     -- not a splitpoint
3468     io.write (line .. "\n") ;
3469 end
3470 end -- do
3471 io.close(sfile)
3472 end -- function
3473
3474
3475 function cvalueerror ( line, linenum , cvalue )
3476 --
3477 -- Incorrect value, so print an error and exit.
3478 --
3479     print ("lwarpmk: ===")
3480     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3481     print (
3482         "lwarpmk: incorrect variable value \"" .. cvalue ..
3483         "\" in lwarpmk.conf.\n"
3484     ) ;
3485     print ("lwarpmk: ===")
3486 --    printconf () ;
```

```
3487     os.exit(1) ;
3488 end
3489
3490
3491 function ignoreconf ()
3492 -- Global argument index
3493 argindex = 2
3494 end
3495
3496 function loadconf ()
3497 --
3498 -- Load settings from the project's "lwarpmk.conf" file:
3499 --
3500 -- Default configuration filename:
3501 local conffile = "lwarpmk.conf"
3502 local confroot = "lwarpmk"
3503 -- Global argument index
3504 argindex = 2
3505 -- Optional configuration filename:
3506 if ( arg[argindex] == "-p" ) then
3507     argindex = argindex + 1
3508     confroot = arg[argindex]
3509     conffile = confroot..".lwarpmkconf"
3510     argindex = argindex + 1
3511 end
3512 -- Additional defaults:
3513 confversion = "0"
3514 opsystem = "Unix"
3515 latexmk = "false"
3516 printlatexcmd = ""
3517 HTMLlatexcmd = ""
3518 printindexcmd = ""
3519 HTMLindexcmd = ""
3520 latexmkindexcmd = ""
3521 -- to be removed:
3522 -- indexprog = "makeindex"
3523 -- makeindexstyle = "lwarp.ist"
3524 -- xindylanguage = "english"
3525 -- xindycodepage = "utf8"
3526 -- xindystyle = "lwarp.xdy"
3527 -- pdftotextenc = "UTF-8"
3528 glossarycmd = "makeglossaries"
3529 -- Verify the file exists:
3530 if (lfs.attributes(conffile,"mode")==nil) then
3531     -- file not exists
3532     print ("lwarpmk: ===")
3533     print ("lwarpmk: File \"" .. conffile .."\" does not exist.")
3534     print ("lwarpmk: Move to the project's source directory,")
3535     print ("lwarpmk: recompile using pdflatex, xelatex, or lualatex,")
3536     print ("lwarpmk: then try using lwarpmk again.")
```

```
3537     if ( arg[argindex] ~= nil ) then
3538         print (
3539             "lwarpmk: (\"" .. confroot ..
3540             "\" does not appear to be a project name.)"
3541         )
3542     end
3543     print ("lwarpmk: ===")
3544     printhelp () ;
3545     os.exit(1) -- exit the entire lwarpmk script
3546 else -- file exists
3547 -- Read the file:
3548 print ("lwarpmk: Reading " .. conffile ..".")
3549 local cfile = io.open(conffile)
3550 -- Scan each line, parsing each line as: name = [[string]]
3551 local linenum = 0
3552 for line in cfile:lines() do -- scan lines
3553 linenum = linenum + 1
3554 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*%[%[([^%]]*)%]%]") ;
3555 -- Error if incorrect enclosing characters:
3556 if ( i == nil ) then
3557     print ("lwarpmk: ===")
3558     print ("lwarpmk: " ..  linenum .. " : " .. line ) ;
3559     print ("lwarpmk: Incorrect entry in " .. conffile ..".\n" ) ;
3560     print ("lwarpmk: ===")
3561 --    printconf () ;
3562     os.exit(1) ;
3563 end -- nil
3564 if ( cvarname == "confversion" ) then
3565     confversion = cvalue
3566 elseif ( cvarname == "opsystem" ) then
3567     -- Verify choice of opsystem:
3568     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
3569         opsystem = cvalue
3570     else
3571         cvalueerror ( line, linenum , cvalue )
3572     end
3573 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
3574 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
3575 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
3576 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
3577 elseif ( cvarname == "printlatexcmd" ) then printlatexcmd = cvalue
3578 elseif ( cvarname == "HTMLlatexcmd" ) then HTMLlatexcmd = cvalue
3579 elseif ( cvarname == "printindexcmd" ) then printindexcmd = cvalue
3580 elseif ( cvarname == "HTMLindexcmd" ) then HTMLindexcmd = cvalue
3581 elseif ( cvarname == "latexmkindexcmd" ) then latexmkindexcmd = cvalue
3582 elseif ( cvarname == "glossarycmd" ) then glossarycmd = cvalue
3583 elseif ( cvarname == "pdftotextenc" ) then pdftotextenc = cvalue
3584 else
3585     print ("lwarpmk: ===")
3586     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
```

```
3587     print (
3588         "lwarpmk: Incorrect variable name \"" .. cvarname .. "\" in " ..
3589         conffile "..\n"
3590     ) ;
3591     print ("lwarpmk: ===")
3592 --    printconf () ;
3593 os.exit(1) ;
3594 end -- cvarname
3595 end -- do scan lines
3596 io.close(cfile)
3597 end -- file exists
3598 -- Error if sourcename is "lwarp".
3599 -- This could happen if a local copy of lwarp has recently been recompiled.
3600 if sourcename=="lwarp" then
3601     print ("lwarpmk: ===")
3602     print ("lwarpmk: lwarp.sty has recently been recompiled in this directory,")
3603     print ("lwarpmk: and \"lwarpmk.conf\" is no longer set for your own project.")
3604     print ("lwarpmk: Recompile your own project using pdf/lua/xelatex <projectname>.")
3605     print ("lwarpmk: After a recompile, \"lwarpmk.conf\" will be set for your project,")
3606     print ("lwarpmk: and you may again use lwarpmk.")
3607     print ("lwarpmk: ===")
3608     os.exit(1)
3609 end -- sourcename of "lwarp"
3610 -- Select some operating-system commands:
3611 if opsystem=="Unix" then  -- For Unix / Linux / Mac OS:
3612     rmname = "rm"
3613     mvname = "mv"
3614     cpname = "cp"
3615     touchnamepre = "touch"
3616     touchnamepost = ""
3617     newtouchname = "touch"
3618     dirslash = "/"
3619     opquote= "\'"
3620     cmdgroupopenname = " ( "
3621     cmdgroupclosename = " ) "
3622     seqname = " && "
3623     bgname = " &"
3624 elseif opsystem=="Windows" then -- For Windows
3625     rmname = "DEL"
3626     mvname = "MOVE"
3627     cpname = "COPY"
3628     touchnamepre = "COPY /b"
3629     touchnamepost = "+,,"
3630     newtouchname = "echo empty >"
3631     dirslash = "\\"
3632     opquote= "\""
3633     cmdgroupopenname = ""
3634     cmdgroupclosename = ""
3635     seqname = " & "
3636     bgname = ""
```

```
else
    print ("lwarpmk: ===")
    print ("lwarpmk: Select Unix or Windows for opsystem." )
    print ("lwarpmk: ===")
    os.exit(1)
end --- for Windows
-- Warning if the operating system does not appear to be correct,
-- in case files were transferred to another system.
if ( (package.config:sub(1,1)) ~= dirslash ) then
    print ("lwarpmk: ===")
    print ("lwarpmk: It appears that lwarpmk.conf is for a different operating system." )
    print ("lwarpmk: To adjust lwarpmk.conf for the current operating system," )
    print ("lwarpmk:   recompile the original document using xe/lua/pdflatex." )
    print ("lwarpmk: ")
    print ("lwarpmk: lwarpmk shall attempt to continue...")
    print ("lwarpmk: ===")
end
-- Error if the configuration file's version is not current:
if ( confversion ~= requiredconfversion ) then
    print ("lwarpmk: ===")
    print ("lwarpmk: The configuration files lwarpmk.conf and "..sourcename..".lwarpmkconf" )
    print ("lwarpmk:   must be updated.  To update the configuration files," )
    print ("lwarpmk:   recompile "..sourcename..".tex using xe/lua/pdflatex," )
    print ("lwarpmk:   then use lwarpmk again.")
    print ("lwarpmk: ===")
    os.exit(1)
end
end -- loadconf


function executecheckerror ( executecommands , errormessage )
--
-- Execute an operating system call,
-- and maybe exit with an error message.
--
local err
err = os.execute ( executecommands )
if ( err ~= 0 ) then
    print ("lwarpmk: ===")
    print ("lwarpmk: " .. errormessage )
    print ("lwarpmk: ===")
    os.exit(1)
end
end -- executecheckerror


function refreshdate ()
os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
end

```

```
3687
3688
3689 function reruntoget (filesource)
3690 --
3691 -- Scan the LaTeX log file for the phrase "Rerun to get",
3692 -- indicating that the file should be compiled again.
3693 -- Return true if found.
3694 --
3695 local fsource = io.open(filesource)
3696 for line in fsource:lines() do
3697 if ( string.find(line,"Rerun to get") ~= nil ) then
3698     io.close(fsource)
3699     return true
3700 end -- if
3701 end -- do
3702 io.close(fsource)
3703 return false
3704 end
3705
3706
3707
3708 function onetime (latexcmd, fsuffix)
3709 --
3710 -- Compile one time, return true if should compile again.
3711 -- fsuffix is "" for print, "_html" for HTML output.
3712 --
3713 print("lwarpmk: Compiling with: " .. latexcmd)
3714 executecheckerror (
3715     latexcmd ,
3716     "Compile error."
3717 )
3718 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
3719 end
3720
3721
3722 function manytimes (latexcmd, fsuffix)
3723 --
3724 -- Compile up to five times.
3725 -- fsuffix is "" for print, "_html" for HTML output
3726 --
3727 if onetime(latexcmd, fsuffix) == true then
3728 if onetime(latexcmd, fsuffix) == true then
3729 if onetime(latexcmd, fsuffix) == true then
3730 if onetime(latexcmd, fsuffix) == true then
3731 if onetime(latexcmd, fsuffix) == true then
3732 end end end end end
3733 end
3734
3735
3736 function verifyfileexists (filename)
```

```
3737 --
3738 -- Exit if the given file does not exist.
3739 --
3740 if (lfs.attributes ( filename , "modification" ) == nil ) then
3741     print ("lwarpmk: ===")
3742     print ("lwarpmk: " .. filename .. " not found." ) ;
3743     print ("lwarpmk: ===")
3744     os.exit (1) ;
3745 end
3746 end
3747
3748
3749
3750 function pdftohtml ()
3751 --
3752 -- Convert <project>_html.pdf into HTML files:
3753 --
3754 -- Convert to text:
3755 print ("lwarpmk: Converting " .. sourcename
3756     .."_html.pdf to " .. sourcename .. "_html.html")
3757 os.execute("pdftotext  -enc " .. pdftotextenc .. "  -nopgbrk  -layout "
3758     .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
3759 -- Split the result into individual HTML files:
3760 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
3761 end
3762
3763
3764 function removeaux ()
3765 --
3766 -- Remove auxiliary files:
3767 -- All .aux files are removed since there may be many bbl*.aux files.
3768 --
3769 os.execute ( rmname .. " *.aux " ..
3770     sourcename ..".toc " .. sourcename .. "_html.toc " ..
3771     sourcename ..".lof " .. sourcename .. "_html.lof " ..
3772     sourcename ..".lot " .. sourcename .. "_html.lot " ..
3773     " *.idx " ..
3774     " *.ind " ..
3775     sourcename ..".ps " .. sourcename .."_html.ps " ..
3776     sourcename ..".log " .. sourcename .. "_html.log " ..
3777     sourcename ..".gl* " .. sourcename .. "_html.gl* " ..
3778     " *_html_inc.* "
3779     )
3780 end
3781
3782 function checkhtmlpdfexists ()
3783 --
3784 -- Error if the HTML document does not exist.
3785 -- The lateximages are drawn from the HTML PDF version of the document,
3786 -- so "lwarpmk html" must be done before "lwarpmk limages".
```

```
3787 --
3788 local htmlpdffile = io.open(sourcename .. "_html.pdf", "r")
3789 if ( htmlpdffile == nil ) then
3790     print ("")
3791     print ("lwarpmk: ===")
3792     print ("lwarpmk: The HTML version of the document does not exist.")
3793     print ("lwarpmk: Enter \"lwarpmk html\" to compile the HTML version.")
3794     print ("lwarpmk: ===")
3795     os.exit(1)
3796 end
3797 io.close (htmlpdffile)
3798 end -- checkhtmlpdfexists
3799
3800
3801 function warnlimages ()
3802 --
3803 -- Warning of a missing lateximages.txt file:
3804 --
3805     print ("lwarpmk: ===")
3806     print ("lwarpmk: \"lateximages.txt\" does not exist.")
3807     print ("lwarpmk: Your project does not use SVG math or other lateximages,")
3808     print ("lwarpmk: or the file has been deleted somehow.")
3809     print ("lwarpmk: Use \"lwarpmk html\" to recompile your project")
3810     print ("lwarpmk: and recreate \"lateximages.txt\".")
3811     print ("lwarpmk: If your project does not use SVG math or other lateximages,")
3812     print ("lwarpmk: then \"lateximages.txt\" will never exist, and")
3813     print ("lwarpmk: \"lwarpmk limages\" will not be necessary.")
3814     print ("lwarpmk: ===")
3815 end -- warnlimages
3816
3817
3818 function checklimages ()
3819 --
3820 -- Check lateximages.txt to see if need to recompile first.
3821 -- If any entry has a page number of zero, then there were incorrect images.
3822 --
3823 print ("lwarpmk: Checking for a valid lateximages.txt file.")
3824 local limagesfile = io.open("lateximages.txt", "r")
3825 if ( limagesfile == nil ) then
3826     warnlimages ()
3827     os.exit(1)
3828 end
3829 -- Track warning to recompile if find a page 0
3830 local pagezerowarning = false
3831 -- Scan lateximages.txt
3832 for line in limagesfile:lines() do
3833     -- lwimgpage is the page number in the PDF which has the image
3834     -- lwimghash is true if this filename is a hash
3835     -- lwimgname is the lateximage filename root to assign for the image
3836     i,j,lwimgpage,lwimghash,lwimgname = string.find (line,"|(.*)|(.*)|(.*)|")
```

```
3837     -- For each entry:
3838     if ( (i~=nil) ) then
3839         -- If the page number is 0, image references are incorrect
3840         --  and must recompile the soure document:
3841         if ( lwimgpage == "0" ) then
3842             pagezerowarning = true
3843         end
3844     end -- if i~=nil
3845 end -- do
3846 if ( pagezerowarning ) then
3847     print ("")
3848     print ("lwarpmk: ===")
3849     print ("lwarpmk: The document must be recompiled before creating the lateximages.")
3850     print ("lwarpmk: Enter \"lwarpmk html\" again, then try \"lwarpmk limages\" again.")
3851     print ("lwarpmk: ===")
3852     os.exit(1) ;
3853 end -- pagezerowarning
3854 end -- checklimages
3855
3856
3857 function createuniximage ( lwimgfullname )
3858 --
3859 -- Create one lateximage for Unix / Linux / Mac OS.
3860 --
3861 executecheckerror (
3862     cmdgroupopenname ..
3863     "pdfseparate -f " .. lwimgpage .. " -l " .. lwimgpage .. " " ..
3864         sourcename .."_html.pdf " ..
3865         "lateximages" .. dirslash .."lateximagetemp-%d" .. ".pdf" ..
3866         seqname ..
3867     -- Crop the image:
3868     "pdfcrop  --hires  lateximages" .. dirslash .. "lateximagetemp-" ..
3869         lwimgpage .. ".pdf " ..
3870         "lateximages" .. dirslash .. lwimgname .. ".pdf" ..
3871         seqname ..
3872     -- Convert the image to svg:
3873     "pdftocairo -svg  -noshrink  lateximages" .. dirslash .. lwimgname .. ".pdf " ..
3874         "lateximages" .. dirslash .. lwimgname ..".svg" ..
3875         seqname ..
3876     -- Remove the temporary files:
3877     rmname .. " lateximages" .. dirslash .. lwimgname .. ".pdf" .. seqname ..
3878     rmname .. " lateximages" .. dirslash .. "lateximagetemp-" .. lwimgpage .. ".pdf" ..
3879     cmdgroupclosename .. " >/dev/null " .. bgname
3880     ,
3881     "File error trying to convert " .. lwimgfullname
3882 )
3883 -- Every 32 images, wait for completion at below normal priority,
3884 --  allowing other image tasks to catch up.
3885 numimageprocesses = numimageprocesses + 1
3886 if ( numimageprocesses > 32 ) then
```

```
3887     numimageprocesses = 0
3888     print ( "lwarpmk: waiting" )
3889     executecheckerror ( "wait" , "File error trying to wait.")
3890 end
3891 end -- createuniximage
3892
3893
3894 function createwindowsimage ( lwimgfullname )
3895 --
3896 -- Create one lateximage for Windows.
3897 --
3898 -- Every 32 images, wait for completion at below normal priority,
3899 --  allowing other image tasks to catch up.
3900 numimageprocesses = numimageprocesses + 1
3901 if ( numimageprocesses > 32 ) then
3902     numimageprocesses = 0
3903     thiswaitcommand = "/WAIT /BELOWNORMAL"
3904     print ( "lwarpmk: waiting" )
3905 else
3906     thiswaitcommand = ""
3907 end
3908 -- Execute the image generation command
3909 executecheckerror (
3910     "start /B " .. thiswaitcommand .. " \"\" lwarp_one_limage " ..
3911     lwimgpage .. " " ..
3912     lwimghash .. " " ..
3913     lwimgname .. " " ..
3914     sourcename .. " <nul >nul"
3915     ,
3916     "File error trying to create image."
3917 )
3918 end -- createwindowsimage
3919
3920
3921 function createonelateximage ( line )
3922 --
3923 -- Given the next line of lateximages.txt, convert a single image.
3924 --
3925 -- lwimgpage is the page number in the PDF which has the image
3926 -- lwimghash is true if this filename is a hash
3927 -- lwimgname is the lateximage filename root to assign for the image
3928 i,j,lwimgpage,lwimghash,lwimgname = string.find (line,"|(.*)|(.*)|(.*)|")
3929 -- For each entry:
3930 if ( (i~=nil) ) then
3931     -- Skip if the page number is 0:
3932     if ( lwimgpage == "0" ) then
3933         pagezerowarning = true
3934     else
3935         -- Skip is this image is hashed and already exists:
3936         local lwimgfullname = "lateximages" .. dirslash .. lwimgname .. ".svg"
```

```
3937        if (
3938            (lwimghash ~= "true") or
3939            (lfs.attributes(lwimgfullname,"mode")==nil) -- file not exists
3940        )
3941        then -- not hashed or not exists:
3942            -- Print the name of the file being generated:
3943            print ( "lwarpmk: " .. lwimgname )
3944            -- Touch/create the dest so that only once instance tries to build it:
3945            executecheckerror (
3946                newtouchname .. " " .. lwimgfullname ,
3947                "File error trying to touch " .. lwimgfullname
3948            )
3949            -- Separate out the image into its own single-page pdf:
3950            if opsystem=="Unix" then
3951                createunixiimage (lwimgfullname)
3952            elseif opsystem=="Windows" then
3953                createwindowsimage (lwimgfullname)
3954            end
3955        end -- not hashed or not exists
3956    end -- not page 0
3957 end -- not nil
3958 end -- createonelateximage
3959
3960
3961 function createlateximages ()
3962 --
3963 -- Create lateximages based on lateximages.txt:
3964 --
3965 -- See if the document must be recompiled first:
3966 checklimages ()
3967 -- See if the print version exists:
3968 checkhtmlpdfexists ()
3969 -- Attempt to create the lateximages:
3970 print ("lwarpmk: Creating lateximages.")
3971 local limagesfile = io.open("lateximages.txt", "r")
3972 if ( limagesfile == nil ) then
3973     warnlateximages ()
3974     os.exit(1)
3975 end
3976 -- Create the lateximages directory, ignore error if already exists
3977 err = os.execute("mkdir lateximages")
3978 -- For Windows, create lwarp_one_limage.cmd from lwarp_one_limage.txt:
3979 if opsystem=="Windows" then
3980     executecheckerror (
3981         cpname .. " lwarp_one_limage.txt lwarp_one_limage.cmd" ,
3982         "File error trying to copy lwarp_one_limage.txt to lwarp_one_limage.cmd"
3983     )
3984 end -- create lwarp_one_limage.cmd
3985 -- Track the number of parallel processes
3986 numimageprocesses = 0
```

```
-- Track warning to recompile if find a page 0
pagezerowarning = false
-- Scan lateximages.txt
for line in limagesfile:lines() do
    createonelateximage ( line )
end -- do
io.close(limagesfile)
print ( "lwarpmk limages: ===")
print ( "lwarpmk limages: Wait a moment for the images to complete" )
print ( "lwarpmk limages:   before reloading the page." )
print ( "lwarpmk limages: ===")
print ( "lwarpmk limages: Done." )
if ( pagezerowarning == true ) then
    print ( "lwarpmk limages: WARNING: Images will be incorrect." )
    print ( "lwarpmk limages:   Enter \"lwarpmk cleanlimages\", then" )
    print ( "lwarpmk limages:   recompile the document one more time, then" )
    print ( "lwarpmk limages:   repeat \"lwarpmk images\" again." )
end -- pagezerowarning
end -- function


function convertepstopdf ()
--
-- Converts EPS files to PDF files.
-- The filenames are arg[argindex] and up.
-- arg[1] is the command "pdftosvg".
--
ignoreconf ()
for i = argindex , #arg do
    if (lfs.attributes(arg[i],"mode")==nil) then
        print ("lwarpmk: File \"" .. arg[i] .. "\" does not exist.")
    else
        print ("lwarpmk: Converting \"" .. arg[i] .. "\"")
        os.execute ( "epstopdf " .. arg[i] )
    end -- if
end -- do
end --function


function convertpdftosvg ()
--
-- Converts PDF files to SVG files.
-- The filenames are arg[argindex] and up.
-- arg[1] is the command "pdftosvg".
--
ignoreconf ()
for i = argindex , #arg do
    if (lfs.attributes(arg[i],"mode")==nil) then
        print ("lwarpmk: File \"" .. arg[i] .. "\" does not exist.")
    else
```

```
4037         print ("lwarpmk: Converting \"" .. arg[i] .. "\"")
4038         os.execute ( "pdftocairo -svg " .. arg[i] )
4039     end -- if
4040 end -- do
4041 end --function
4042
4043
4044 -- Force an update and conclude processing:
4045 function updateanddone ()
4046 print ("lwarpmk: Forcing an update of " .. sourcename ..".tex.")
4047 refreshdate ()
4048 print ("lwarpmk: " .. sourcename ..".tex is ready to be recompiled.")
4049 print ("lwarpmk: Done.")
4050 end -- function
4051
4052
4053 -- Start of the main code: --
4054
4055
4056 -- lwarpmk --version :
4057
4058 if (arg[1] == "--version") then
4059 print ( "lwarpmk: " .. printversion )
4060
4061 else -- not --version
4062
4063
4064 -- print intro:
4065
4066 print ("lwarpmk: " .. printversion .. "  Automated make for the LaTeX lwarp package.")
4067
4068
4069 -- lwarpmk print:
4070
4071 if arg[1] == "print" then
4072 loadconf ()
4073 if ( latexmk == "true" ) then
4074     print ("lwarpmk: Compiling with: " .. printlatexcmd)
4075     executecheckerror (
4076         printlatexcmd ,
4077         "Compile error."
4078     )
4079     print ("lwarpmk: Done.")
4080 else -- not latexmk
4081     verifyfileexists (sourcename .. ".tex") ;
4082     -- See if up to date:
4083     if (
4084         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
4085         (
4086             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
```

```
4087             lfs.attributes ( sourcename .. ".pdf" , "modification" )
4088         )
4089     ) then
4090         -- Recompile if not yet up to date:
4091         manytimes(printlatexcmd, "")
4092         print ("lwarpmk: Done.") ;
4093     else
4094         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
4095     end
4096 end -- not latexmk
4097
4098
4099 -- lwarpmk print1:
4100
4101 elseif arg[1] == "print1" then
4102     loadconf ()
4103     verifyfileexists (sourcename .. ".tex") ;
4104     onetime(printlatexcmd, "")
4105     print ("lwarpmk: Done.") ;
4106
4107
4108 -- lwarpmk printindex:
4109 -- Compile the index then touch the source
4110 -- to trigger a recompile of the document:
4111
4112 elseif arg[1] == "printindex" then
4113 loadconf ()
4114 os.execute ( printindexcmd )
4115 print ("lwarpmk: -------")
4116 updateanddone ()
4117
4118
4119 -- lwarpmk printglossary:
4120 -- Compile the glossary then touch the source
4121 -- to trigger a recompile of the document:
4122
4123 elseif arg[1] == "printglossary" then
4124 loadconf ()
4125 print ("lwarpmk: Processing the glossary.")
4126
4127 os.execute(glossarycmd .. " " .. sourcename)
4128 updateanddone ()
4129
4130
4131 -- lwarpmk html:
4132
4133 elseif arg[1] == "html" then
4134 loadconf ()
4135 if ( latexmk == "true" ) then
4136     print ("lwarpmk: Compiling with: " .. HTMLlatexcmd)
```

```
4137     executecheckerror (
4138         HTMLlatexcmd ,
4139         "Compile error."
4140     )
4141     pdftohtml ()
4142     print ("lwarpmk: Done.")
4143 else -- not latexmk
4144     verifyfileexists ( sourcename .. ".tex" ) ;
4145     -- See if exists and is up to date:
4146     if (
4147         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
4148         (
4149             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
4150             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
4151         )
4152     ) then
4153         -- Recompile if not yet up to date:
4154         manytimes(HTMLlatexcmd, "_html")
4155         pdftohtml ()
4156         print ("lwarpmk: Done.")
4157     else
4158         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
4159     end
4160 end -- not latexmk
4161
4162
4163 -- lwarpmk html1:
4164
4165 elseif arg[1] == "html1" then
4166     loadconf ()
4167     verifyfileexists ( sourcename .. ".tex" ) ;
4168     onetime(HTMLlatexcmd, "_html")
4169     pdftohtml ()
4170     print ("lwarpmk: Done.")
4171
4172
4173 -- lwarpmk pdftohtml:
4174 elseif arg[1] == "pdftohtml" then
4175     loadconf ()
4176     pdftohtml ()
4177
4178
4179 -- lwarpmk htmlindex:
4180 -- Compile the index then touch the source
4181 -- to trigger a recompile of the document:
4182
4183 elseif arg[1] == "htmlindex" then
4184 loadconf ()
4185 os.execute ( HTMLindexcmd )
4186 print ("lwarpmk: -------")
```

```
updateanddone ()


-- lwarpmk htmlglossary:
-- Compile the glossary then touch the source
-- to trigger a recompile of the document.
-- The <sourcename>.xdy file is created by the glossaries package.

elseif arg[1] == "htmlglossary" then
loadconf ()
print ("lwarpmk: Processing the glossary.")
os.execute(glossarycmd .. " " .. sourcename .. "_html")
updateanddone ()


-- lwarpmk limages:
-- Scan the lateximages.txt file to create lateximages.

elseif arg[1] == "limages" then
loadconf ()
print ("lwarpmk: Processing images.")
createlateximages ()
print ("lwarpmk: Done.")


-- lwarpmk again:
-- Touch the source to trigger a recompile.

elseif arg[1] == "again" then
loadconf ()
updateanddone ()


-- lwarpmk clean:
-- Remove project.aux, .toc, .lof, .lot, .log, *.idx, *.ind, *_html_inc.*, .gl*

elseif arg[1] == "clean" then
loadconf ()
removeaux ()
print ("lwarpmk: Done.")


-- lwarpmk cleanall
-- Remove project.aux, .toc, .lof, .lot, .log, *.idx, *.ind, *_html_inc.*, .gl*
--    and also project.pdf, project.dvi, *.html

elseif arg[1] == "cleanall" then
loadconf ()
removeaux ()
os.execute ( rmname .. " " ..
```

```
4237     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
4238     sourcename .. ".dvi " .. sourcename .. "_html.dvi " ..
4239     "*.html"
4240     )
4241 print ("lwarpmk: Done.")
4242
4243
4244 -- lwarpmk cleanlimages
4245 -- Remove images from the lateximages directory.
4246
4247 elseif arg[1] == "cleanlimages" then
4248 loadconf ()
4249 os.execute ( rmname .. " lateximages/*" )
4250 print ("lwarpmk: Done.")
4251
4252 -- lwarpmk epstopdf <list of file names>
4253 -- Convert EPS files to PDF using epstopdf
4254 elseif arg[1] == "epstopdf" then
4255 convertepstopdf ()
4256 print ("lwarpmk: Done.")
4257
4258
4259 -- lwarpmk pdftosvg <list of file names>
4260 -- Convert PDF files to SVG using pdftocairo
4261 elseif arg[1] == "pdftosvg" then
4262 convertpdftosvg ()
4263 print ("lwarpmk: Done.")
4264
4265
4266 -- lwarpmk with no argument :
4267
4268 elseif (arg[1] == nil) then
4269 printhelp ()
4270
4271
4272 -- lwarpmk -h or lwarpmk --help :
4273
4274 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
4275 printusage ()
4276
4277
4278 -- Unknown command:
4279
4280 else
4281 printhelp ()
4282 print ("\nlwarpmk: ****** Unknown command \""..arg[1].."\". ******\n")
4283 end
4284
4285 end -- not --version
4286 \end{filecontents*}
```

```
% \end{Verbatim}% for syntax highlighting

\end{LWR@createlwarpmk}
```

38 Stacks

for HTML output:

```
\begin{warpHTML}
```

Stacks are used to remember how to close sections and list items. Before a new section is started, previously nested sections and items must be closed out (un-nested) in proper order. Note that starting a new section may close several levels of previously nested items at the same time. For example, starting a new `\section` would close any currently open subsection, subsubsection, and paragraph. General environments are not nested on the stack since they have their own close mechanism. List environments are nested, and items inside those environments are nested one level deeper still. List environments may be nested inside other list environments, and list items are nested inside list environments as well. Thus, the stack may have items which are not necessarily in order, since a description may contain an enumerate, for example. Depths to be recorded in `\LWR@closedepthone`, etc.

38.1 Assigning depths

initial depths for empty stack entries:

```
\newcommand*{\LWR@depthnone}{-5}
```

all sectioning depths are deeper than `LWR@depthfinished`:

```
\newcommand*{\LWR@depthfinished}{-4}
\newcommand*{\LWR@depthpart}{-1}
\newcommand*{\LWR@depthchapter}{0}
\newcommand*{\LWR@depthsection}{1}
\newcommand*{\LWR@depthsubsection}{2}
\newcommand*{\LWR@depthsubsubsection}{3}
\newcommand*{\LWR@depthparagraph}{4}
\newcommand*{\LWR@depthsubparagraph}{5}
```

used by `\itemize`, `\enumerate`, `\description`:

```
\newcommand*{\LWR@depthlist}{6}
```

used by `\item`:

```
\newcommand*{\LWR@depthlistitem}{7}
```

38.2 Closing actions

A stack to record the action to take to close each nesting level: Add more levels of stack if necessary for a very deeply nested document, adding to `\pushclose` and `\popclose` as well.

```
\newcommand*{\LWR@closeone}{}% top of the stack
\newcommand*{\LWR@closetwo}{}
\newcommand*{\LWR@closethree}{}
\newcommand*{\LWR@closefour}{}
\newcommand*{\LWR@closefive}{}
\newcommand*{\LWR@closesix}{}
\newcommand*{\LWR@closeseven}{}
\newcommand*{\LWR@closeeight}{}
\newcommand*{\LWR@closenine}{}
\newcommand*{\LWR@closeten}{}
\newcommand*{\LWR@closeeleven}{}
\newcommand*{\LWR@closetwelve}{}
```

38.3 Closing depths

A stack to record the depth of each level:

⚠ Note that nested LaTeX structures may push depths which are non-sequential.

Ex:

```
\begin{itemize}
   \item{A}
   \begin{description}
      \item{B}
   \end{description}
\end{itemize}
```

```
\newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
\newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
\newcommand*{\LWR@closedepththree}{\LWR@depthnone}
\newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
\newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
\newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
```

```
\newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
\newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
\newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
\newcommand*{\LWR@closedepththen}{\LWR@depthnone}
\newcommand*{\LWR@closedepththeleven}{\LWR@depthnone}
\newcommand*{\LWR@closedepththtwelve}{\LWR@depthnone}
```

38.4 Pushing and popping the stack

`\pushclose` {⟨*action*⟩} {⟨*depth*⟩}

Pushes one return action and its LaTeX depth onto the stacks.

```
\NewDocumentCommand{\pushclose}{m m}
{
\global\let\LWR@closetwelve\LWR@closeeleven
\global\let\LWR@closeeleven\LWR@closeten
\global\let\LWR@closeten\LWR@closenine
\global\let\LWR@closenine\LWR@closeeight
\global\let\LWR@closeeight\LWR@closeseven
\global\let\LWR@closeseven\LWR@closesix
\global\let\LWR@closesix\LWR@closefive
\global\let\LWR@closefive\LWR@closefour
\global\let\LWR@closefour\LWR@closethree
\global\let\LWR@closethree\LWR@closetwo
\global\let\LWR@closetwo\LWR@closeone
\global\let\LWR@closeone#1
\global\let\LWR@closedepththtwelve\LWR@closedepththeleven
\global\let\LWR@closedepththeleven\LWR@closedepththen
\global\let\LWR@closedepththen\LWR@closedepthnine
\global\let\LWR@closedepthnine\LWR@closedeptheight
\global\let\LWR@closedeptheight\LWR@closedepthseven
\global\let\LWR@closedepthseven\LWR@closedepthsix
\global\let\LWR@closedepthsix\LWR@closedepthfive
\global\let\LWR@closedepthfive\LWR@closedepthfour
\global\let\LWR@closedepthfour\LWR@closedepththree
\global\let\LWR@closedepththree\LWR@closedepthtwo
\global\let\LWR@closedepthtwo\LWR@closedepthone
\global\let\LWR@closedepthone#2
}
```

`\popclose` Pops one action and its depth off the stacks.

```
\newcommand*{\popclose}
{
\global\let\LWR@closeone\LWR@closetwo
```

```
\global\let\LWR@closetwo\LWR@closethree
\global\let\LWR@closethree\LWR@closefour
\global\let\LWR@closefour\LWR@closefive
\global\let\LWR@closefive\LWR@closesix
\global\let\LWR@closesix\LWR@closeseven
\global\let\LWR@closeseven\LWR@closeeight
\global\let\LWR@closeeight\LWR@closenine
\global\let\LWR@closenine\LWR@closeten
\global\let\LWR@closeten\LWR@closeeleven
\global\let\LWR@closeeleven\LWR@closetwelve
\global\let\LWR@closedepthone\LWR@closedepththwo
\global\let\LWR@closedepththwo\LWR@closedepththree
\global\let\LWR@closedepththree\LWR@closedepthfour
\global\let\LWR@closedepthfour\LWR@closedepthfive
\global\let\LWR@closedepthfive\LWR@closedepthsix
\global\let\LWR@closedepthsix\LWR@closedepthseven
\global\let\LWR@closedepthseven\LWR@closedeptheight
\global\let\LWR@closedeptheight\LWR@closedepthnine
\global\let\LWR@closedepthnine\LWR@closedepthten
\global\let\LWR@closedepthten\LWR@closedeptheleven
\global\let\LWR@closedeptheleven\LWR@closedepththwelve
}

\end{warpHTML}
```

39 Data arrays

These macros are similar to the **arrayjobx** package, except that `\LWR@setexparray`'s argument is expanded only once when assigned.

`name` has no backslash, `index` can be a number or a text name, and an empty `value` must be `\relax` instead of empty.

To assign an empty value:

```
\LWR@setexparray{name}{index}{}
```

for HTML output:

```
\begin{warpHTML}
```

`\LWR@setexparray` `{`⟨*name*⟩`}` `{`⟨*index*⟩`}` `{`⟨*contents*⟩`}`

```
\NewDocumentCommand{\LWR@setexparray}{m m m}{%
    \xdef\LWR@thisexparrayname{#1#2}%
    \ifstrempty{#3}%
    {\csgdef{\LWR@thisexparrayname}{}}%
    {\csxdef{\LWR@thisexparrayname}{#3}}%
```

```
}
```

`\LWR@getexparray` {⟨*name*⟩} {⟨*index*⟩}

```
\newcommand*{\LWR@getexparray}[2]{%
    \@nameuse{#1#2}%
}

\end{warpHTML}
```

40 Localizing catcodes

for HTML & PRINT:

```
\begin{warpall}
```

⚠ misplaced alignment tab character &

Place `\StartDefiningTabulars` and `\StopDefiningTabulars` before and after defining macros or environments which include the tabular `&` character in their definitions.

The catcode of `&` must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

```
misplaced alignment tab character &
```

`\StartDefiningTabulars` Place before defining something with `&` in it.

```
\newcommand{\StartDefiningTabulars}{%
\LWR@traceinfo{StartDefiningTabulars}%
\warpHTMLonly{\catcode`\&=\active}%
}
```

`\StopDefiningTabulars` Place after defining something with `&` in it.

```
\newcommand{\StopDefiningTabulars}{%
\LWR@traceinfo{StopDefiningTabulars}%
\warpHTMLonly{\catcode`\&=4}%
}
```

Bool `LWR@mathmacro` True if currently defining math macros. Used to disable SVG math hashing and MathJax math contents while defining a macro using inline math. Begin a macro, it is not guaranteed that the contents are static, and so the image must be unique. The contents also almost certainly will not be parsed correctly by MathJax.

```
\newbool{LWR@mathmacro}
\boolfalse{LWR@mathmacro}
```

\StartDefiningMath Place before defining something with $ in it.

```
\newcommand{\StartDefiningMath}{%
\LWR@traceinfo{StartDefiningMath}%
\warpHTMLonly{\catcode`\$=\active}%
}
```

\StopDefiningMath Place after defining something with $ in it.

```
\newcommand{\StopDefiningMath}{%
\LWR@traceinfo{StopDefiningMath}%
\warpHTMLonly{\catcode`\$=3}% math shift
}
```

```
\end{warpall}
```

41 Localizing dynamic math

Inline SVG math usually uses a hash of its contents to generate `lateximages` which are reusable for multiple instances with the same contents. If the contents may change for each use, such as depending on the current value of a counter, then `\StartDynamicMath` must be used before the inline math expression, and `\StopDynamicMath` must be used after.

For MathJax, the inline math expression is usually printed for MathJax to interpret. When marked as dynamic math, the following inline math expression will be displayed as an unhashed inline SVG image instead.

For existing code and packages, it may be possible to patch macros after they have been defined, using the **xpatch** package, which is pre-loaded by **lwarp**:

```
\xpatchcmd{\macroname}
    {$math expression$}
    {\StartDynamicMath$math expression$\StopDynamicMath}
    {}
    {\typeout{Error patching macroname.}}
```

for HTML & PRINT:

```
\begin{warpall}
```

Bool `LWR@dynamicmath` True to mark inline math which is dynamic in nature, thus should not be hashed for reuse.
Default: `false`

```
\newbool{LWR@dynamicmath}
\boolfalse{LWR@dynamicmath}
```

\StartDynamicMath Place before using $ … $ or \(…\) if the contents of the math are not static, depending on counters or dynamic macros.

```
\newcommand{\StartDynamicMath}{%
\LWR@traceinfo{StartDynamicMath}%
\booltrue{LWR@dynamicmath}%
}
```

\StopDynamicMath Place after using $ … $ or \(…\) with dynamic contents.

```
\newcommand{\StopDynamicMath}{%
\LWR@traceinfo{StopDynamicMath}%
\boolfalse{LWR@dynamicmath}%
}
```

```
\end{warpall}
```

42 Sanitizing labels and filenames

Special handling for underscores in labels and filenames.

for HTML output:

```
\begin{warpHTML}
```

\LWR@sanitized The sanitized version of what was given to \LWR@sanitize. Characters are set to their detokenized versions. Required for underscores in labels and filenames.

```
\newcommand*{\LWR@sanitized}{}
```

\LWR@sanitize {⟨*text*⟩}

Sanitizes the text and returns the result in \LWR@sanitized.

```
\newcommand*{\LWR@sanitize}[1]{%
\LWR@traceinfo{LWR@sanitize: !#1!}%
\edef\LWR@sanitized{#1}%
\LWR@traceinfo{LWR@sanitize expanded: !\LWR@sanitized!}%
\edef\LWR@sanitized{\detokenize\expandafter{\LWR@sanitized}}%
\LWR@traceinfo{LWR@sanitize result: !\LWR@sanitized!}%
}
```

```
\end{warpHTML}
```

43 HTML entities

for HTML output:

```
\begin{warpHTML}
```

HTML entites and HTML Unicode entities:

```
\let\LWR@origampersand\&
```

`\HTMLentity` {⟨*entitytag*⟩}

```
\newcommand*{\HTMLentity}[1]{%
% \LWR@traceinfo{HTMLentity \detokenize{#1}}%
\begingroup%
\LWR@FBcancel%
\LWR@origampersand#1;%
\endgroup
% \LWR@traceinfo{HTMLentity done}%
}
```

`\HTMLunicode` {⟨*hex_unicode*⟩}

```
\newcommand*{\HTMLunicode}[1]{\HTMLentity{\LWR@origpound{}x#1}}
```

`\&`

```
\renewrobustcmd*{\&}{\HTMLentity{amp}}
```

`\textless`
`\textgreater`

```
\let\LWR@origtextless\textless
\renewcommand*{\textless}{\HTMLentity{lt}}

\let\LWR@origtextgreater\textgreater
\renewcommand*{\textgreater}{\HTMLentity{gt}}
```

```
\end{warpHTML}
```

44 HTML filename generation

The filename of the homepage is set to `\HomeHTMLFilename.html`. The filenames of additional sections start with `\HTMLFilename`, to which is appended a section number or a simplified section name, depending on `FileSectionNames`.

for HTML & PRINT:

```
\begin{warpall}
```

`\BaseJobname` The `\jobname` of the printed version, even if currently compiling the HTML version. I.e. this is the `\jobname` without `_html` appended. This is used to set `\HomeHTMLFilename` if the user did not provide one.

```
\providecommand*{\BaseJobname}{\jobname}
```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 8.4.1.

```
\providecommand*{\HTMLFilename}{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 8.4.1.

```
\providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

`\SetHTMLFileNumber` `{`⟨*number*⟩`}`

Sets the file number for the next file to be generated. 0 is the home page. Use just before the next sectioning command, and set it to one less than the desired number of the next section. May be used to generate numbered groups of nodes such as 100+ for one chapter, 200+ for another chapter, etc.

```
\newcommand*{\SetHTMLFileNumber}[1]{%
\setcounter{LWR@htmlfilenumber}{#1}%
}
```

Bool `FileSectionNames` Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
\newbool{FileSectionNames}
\booltrue{FileSectionNames}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Ctr `LWR@htmlfilenumber` Records the number of each HTML file as it is being created. Number 0 is the home page.

```
\newcounter{LWR@htmlfilenumber}
\setcounter{LWR@htmlfilenumber}{0}
```

\LWR@htmlsectionfilename {⟨*htmlfilenumber or name*⟩}

Prints the filename for a given section: \HTMLFilename{}filenumber/name.html

```
\newcommand*{\LWR@htmlsectionfilename}[1]{%
\LWR@traceinfo{LWR@htmlsectionfilename A !\detokenize{#1}!}%
```

Section 0 or empty is given the home filename. The filename must be detokenized for underscores.

```
% \LWR@traceinfo{about to assign temp}%
\edef\LWR@tempone{#1}%
\LWR@traceinfo{about to compare with ??}%
\ifthenelse{\equal{\LWR@tempone}{??}}%
{\LWR@traceinfo{found ??}}%
{\LWR@traceinfo{not found ??}}%
\LWR@traceinfo{about to compare with zero or empty}%
\ifthenelse{%
    \equal{\LWR@tempone}{0}%
    \OR \equal{\LWR@tempone}{}%
    \OR \equal{\LWR@tempone}{??}%
}%
{%
    \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
    \HomeHTMLFilename.html%
}%
```

For a LaTeX section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename index.html:

```
{%
    \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
    \ifthenelse{%
        \equal{\HTMLFilename}{} \AND
        \equal{\LWR@tempone}{Index} \OR
        \equal{\LWR@tempone}{index}%
    }%
    {%
        \LWR@traceinfo{Prefixing the index name with an underscore.}%
        \_#1.html%
    }%
```

Otherwise, create a filename with the chosen prefix:

```
    {\HTMLFilename#1.html}%
}%
\LWR@traceinfo{LWR@htmlsectionfilename Z}%
}
```

\LWR@htmlrefsectionfilename {⟨*label*⟩}

Prints the filename for the given label

```
\newcommand*{\LWR@htmlrefsectionfilename}[1]{%
\LWR@traceinfo{LWR@htmlrefsectionfilename: !\detokenize{#1}!}%
```

\LWR@nullfonts to allow math in a section name.

```
\begingroup%
\LWR@nullfonts%
\LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
\endgroup%
\LWR@traceinfo{LWR@htmlrefsectionfilename: done}%
}
```

```
\end{warpHTML}
```

45 Homepage link

for HTML output:

```
\begin{warpHTML}
```

\LinkHome May be used wherever you wish to place a link back to the homepage. The filename must be detokenized for underscores.

```
\newcommand*{\LinkHome}{%
\LWR@subhyperrefclass{\HomeHTMLFilename.html}{Home}{linkhome}%
}
```

\LWR@topnavigation Creates a link to the homepage at the top of the page for use when the window is too narrow for the sideTOC.

```
\newcommand*{\LWR@topnavigation}{
\LWR@htmlelementclassline{nav}{topnavigation}{\LinkHome}
}
```

\LWR@botnavigation Creates a link to the homepage at the bottom of the page for use when the window is too narrow for the sideTOC.

```
\newcommand*{\LWR@botnavigation}{
\LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}
}
```

```
\end{warpHTML}
```

46 \LWRPrintStack diagnostic tool

⚠ Diagnostics tool: Prints the LaTeX nesting depth values for the stack levels. \LWR@startpars is used before printing the stack, so that \LWRPrintStack may be called from anywhere in the normal text flow.

for HTML output:
```
\begin{warpHTML}
```

\LWRPrintStack Prints the closedepth stack.

```
\newcommand*{\LWR@subprintstack}{
\LWR@closedepthone\ \LWR@closedepthtwo\ \LWR@closedepththree\
\LWR@closedepthfour\ \LWR@closedepthfive\ \LWR@closedepthsix\
\LWR@closedepthseven\ \LWR@closedeptheight\ \LWR@closedepthnine\
\LWR@closedepthten\ \LWR@closedepthteleven\ \LWR@closedepthtwelve\
}

\newcommand*{\LWRPrintStack}{
\LWR@startpars
\LWR@subprintstack
}

\end{warpHTML}
```

for PRINT output:
```
\begin{warpprint}

\newcommand*{\LWRPrintStack}{}

\end{warpprint}
```

47 Closing stack levels

for HTML output:
```
\begin{warpHTML}
```

Close one nested level:

```
\newcommand*{\LWR@closeoneprevious}{%

\LWR@closeone

\popclose
}
```

\LWR@closeprevious {⟨*depth*⟩} Close everything up to the given depth:

```
\newcommand*{\LWR@closeprevious}[1]{
\LWR@traceinfo{LWR@closeprevious to depth #1, depths are \LWR@subprintstack}%
```

Close any pending paragraph:

```
\LWR@stoppars%
```

Close anything nested deeper than the desired depth. First close anything deeper, then at most one of the same level.

```
\whileboolexpr{test{\ifnumcomp{\LWR@closedepthone}{>}{#1}}}%
{%
    \LWR@traceinfo{LWR@closeprevious: closing out depth \LWR@closedepthone}%
    \LWR@closeoneprevious%
}%
\ifboolexpr{test{\ifnumcomp{\LWR@closedepthone}{=}{#1}}}%
{%
    \LWR@traceinfo{LWR@closeprevious: closing out depth \LWR@closedepthone}%
    \LWR@closeoneprevious%
}{}%
\LWR@traceinfo{LWR@closeprevious: done, depths are \LWR@subprintstack}%
}
```

```
\end{warpHTML}
```

48 PDF pages and styles

for HTML output:

```
\begin{warpHTML}
```

\LWR@forcenewpage New PDF page a before major environment.

This is used just before major environments, such as `verse`. Reduces the chance of an environment overflowing the HTML PDF output page.

```
\newcommand{\LWR@forcenewpage}{%
\LWR@traceinfo{LWR@forcenewpage}%
\ifinner\else%
\LWR@stoppars\LWR@orignewpage\LWR@startpars%
\fi%
}
```

`\pagestyle`, etc. are nullified for HTML output.

\pagestyle {⟨*style*⟩}

```
\renewcommand*{\pagestyle}[1]{}
```

\thispagestyle {⟨*style*⟩}

```
\renewcommand*{\thispagestyle}[1]{}
```

\markboth {⟨*left*⟩} {⟨*right*⟩}

```
\renewcommand*{\markboth}[2]{}
```

\markright {⟨*right*⟩}

```
\renewcommand*{\markright}[1]{}
```

\raggedbottom

```
\renewcommand*{\raggedbottom}{}
```

\flushbottom

```
\renewcommand*{\flushbottom}{}
```

\sloppy

```
\renewcommand*{\sloppy}{}
```

\fussy

```
\renewcommand*{\fussy}{}
```

\pagenumbering * {⟨*commands*⟩}

```
\RenewDocumentCommand{\pagenumbering}{s m}{}
\end{warpHTML}
```

49 HTML tags, spans, divs, elements

for HTML output:

```
\begin{warpHTML}
```

49.1 Mapping LaTeX Sections to HTML Sections

```
\newcommand*{\LWR@tagtitle}{h1}
\newcommand*{\LWR@tagtitleend}{/h1}
\newcommand*{\LWR@tagpart}{h2}
\newcommand*{\LWR@tagpartend}{/h2}
\newcommand*{\LWR@tagchapter}{h3}
\newcommand*{\LWR@tagchapterend}{/h3}
\newcommand*{\LWR@tagsection}{h4}
\newcommand*{\LWR@tagsectionend}{/h4}
\newcommand*{\LWR@tagsubsection}{h5}
\newcommand*{\LWR@tagsubsectionend}{/h5}
\newcommand*{\LWR@tagsubsubsection}{h6}
\newcommand*{\LWR@tagsubsubsectionend}{/h6}
\newcommand*{\LWR@tagparagraph}{span class="paragraph"}
\newcommand*{\LWR@tagparagraphend}{/span}
\newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
\newcommand*{\LWR@tagsubparagraphend}{/span}

\newcommand*{\LWR@tagregularparagraph}{p}
```

49.2 Babel-French tag modifications

Adjust **babel-french** for HTML spaces. So far, this only works for **pdflatex** and **xelatex**.

(*Emulates or patches code by* Daniel Flipo.)

```
\providecommand*{\LWR@FBcancel}{}

\AtBeginDocument{%
\@ifundefined{frenchbsetup}%
{}%
{%
    \frenchbsetup{FrenchFootnotes=false}%

%
    \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
    \renewrobustcmd*{\FBcolonspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;%
        \endgroup%
    }%
    \renewrobustcmd*{\FBthinspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand\LWR@origpound{}x202f;% \,
        \endgroup%
```

```
    }%
    \renewrobustcmd*{\FBguillspace}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;% ~,  for \og xyz \fg{}
        \endgroup%
    }%
    \DeclareDocumentCommand{\FBmedkern}{}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand\LWR@origpound{}x202f;% \,
        \endgroup%
    }%
    \DeclareDocumentCommand{\FBthickkern}{}{%
        \begingroup%
        \LWR@FBcancel%
        \LWR@origampersand{}nbsp;% ~
        \endgroup%
    }%
    \renewrobustcmd*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
    \ifFBunicode%
    \else%
        \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
        \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
    \fi%
}%
}
```

49.3 HTML tags

`\LWR@htmltagc` {⟨*tag*⟩} Break ligatures and use upright apostrophes in HTML tags.

`\protect` is in case the tag appears in TOC, LOF, LOT.

```
\newcommand*{\LWR@htmltagc}[1]{%
\LWR@traceinfo{LWR@htmltagc !\detokenize{#1}!}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
\protect\LWR@origtextless%
#1%
\protect\LWR@origtextgreater%
\endgroup%
% \LWR@traceinfo{LWR@htmltagc: done}%
}
```

Env `LWR@nestspan` Disable `minipage`, `\parbox`, and HTML `<div>`s inside a `<span>`.

⚠ \begin{LWR@nestspan} must follow the opening tag to allow a paragraph to start if the span is at the beginning of a new paragraph.

⚠ \end{LWR@nestspan} must follow the or a <p> may appear inside the span.

```
\newcommand*{\LWR@nestspanitem}{%
\if@newlist\else{\LWR@htmltagc{br /}}\fi%
\LWR@origitem%
}

\newenvironment*{LWR@nestspan}
{%
\LWR@traceinfo{LWR@nestspan starting}%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{%
    \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
}%
{% not in a lateximage
    \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
    \addtocounter{LWR@spandepth}{1}%
    \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{}%
    \RenewDocumentEnvironment{BlockClass}{o m}{}{}%
    \renewcommand{\BlockClassSingle}[2]{##2}%
    \renewcommand{\LWR@forcenewpage}{}%
    \renewcommand{\LWR@liststart}{%
        \let\item\LWR@nestspanitem%
    }%
    \renewcommand{\LWR@listend}{\LWR@htmltagc{br /}\LWR@htmltagc{br /}}%
}% not in a lateximage
\LWR@traceinfo{LWR@nestspan starting: done}%
}% starting env
{% ending env
\LWR@traceinfo{LWR@nestspan ending}%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{}%
{\addtocounter{LWR@spandepth}{-1}}%
\LWR@traceinfo{LWR@nestspan ending: done}%
}

\AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}
```

\LWR@htmlspan {⟨*tag*⟩} {⟨*text*⟩}

⚠ \LWR@spandepth is used to ensure that paragraph tags are not generated inside a span. The exact sequence of when to add and subtract the counter is important to correctly handle the paragraph tags before and after the span.

```
\NewDocumentCommand{\LWR@htmlspan}{m +m}{%
\LWR@ensuredoingapar%
```

```
\LWR@htmltagc{#1}%
\begin{LWR@nestspan}%
#2%
\LWR@htmltagc{/#1}%
\end{LWR@nestspan}%
}
```

`\LWR@htmlspanclass` `[`⟨*style*⟩`]` `{`⟨*class*⟩`}` `{`⟨*text*⟩`}`

```
\NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
\LWR@traceinfo{LWR@htmlspanclass |#1|#2|}%
\LWR@ensuredoingapar%
\LWR@subhtmlelementclass{span}[#1]{#2}%
\begin{LWR@nestspan}%
#3%
\LWR@htmltagc{/span}%
\LWR@traceinfo{LWR@htmlspanclass done}%
\end{LWR@nestspan}%
}
```

`\LWR@htmltag` `{`⟨*tag*⟩`}`

Print an HTML tag: <tag>

```
\newcommand*{\LWR@htmltag}[1]{%
% \LWR@traceinfo{LWR@htmltagb !\detokenize{#1}!}%
\LWR@htmltagc{#1}%
% \LWR@traceinfo{LWR@htmltagb: done}%
}
```

49.4 Block tags and comments

In the following, `\origttfamily` breaks ligatures, which may not be used for HTML codes:

`\LWR@htmlopencomment`
`\LWR@htmlclosecomment`

```
\newcommand*{\LWR@htmlopencomment}{%
{%
% \LWR@traceinfo{LWR@htmlopencomment}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
\LWR@print@mbox{\LWR@origtextless{}!{-}{-}}%
```

```
\endgroup%
}%
}

\newcommand*{\LWR@htmlclosecomment}{%
{%
% \LWR@traceinfo{LWR@htmlclosecomment}%
\begingroup%
\LWR@FBcancel%
\ifmmode\else\protect\LWR@origttfamily\fi%
\LWR@print@mbox{{-}{-}\LWR@origtextgreater}%
\endgroup%
}%
}
```

`\LWR@htmlcomment` {⟨*comment*⟩}

```
\newcommand{\LWR@htmlcomment}[1]{%
\LWR@htmlopencomment{}%
{%
\LWR@origttfamily% break ligatures
#1%
}%
\LWR@htmlclosecomment{}}
```

`\LWR@htmlblockcomment` {⟨*comment*⟩}

```
\newcommand{\LWR@htmlblockcomment}[1]
{\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars}
```

`\LWR@htmlblocktag` {⟨*tag*⟩} print a stand-alone HTML tag

```
\newcommand*{\LWR@htmlblocktag}[1]{%
\LWR@stoppars%
\LWR@htmltag{#1}%
\LWR@startpars%
}
```

49.5 Div class and element class

`\LWR@subhtmlelementclass` {⟨*element*⟩} [⟨*style*⟩] {⟨*class*⟩}

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
\NewDocumentCommand{\LWR@subhtmlelementclass}{m O{} m}{%
\LWR@traceinfo{LWR@subhtmlelementclass !#1!#2!#3!}%
\ifblank{#2}%
{\LWR@htmltag{#1 class="#3"}}% empty option
{\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
\LWR@traceinfo{LWR@subhtmlelementclass done}%
}
```

`\LWR@htmlelementclass` {⟨*element*⟩} {⟨*class*⟩} [⟨*style*⟩]

```
\NewDocumentCommand{\LWR@htmlelementclass}{m o m}{%
\LWR@stoppars%
\LWR@subhtmlelementclass{#1}[#2]{#3}%
\LWR@startpars%
}
```

`\LWR@htmlelementclassend` {⟨*element*⟩} {⟨*class*⟩}

```
\newcommand*{\LWR@htmlelementclassend}[2]{%
\LWR@stoppars%
\LWR@htmltag{/#1}%
\ifbool{HTMLDebugComments}{%
    \LWR@htmlcomment{End of #1 ``#2''}%
}{}%
\LWR@startpars%
}
```

`\LWR@htmldivclass` [⟨*style*⟩] {⟨*class*⟩}

```
\NewDocumentCommand{\LWR@htmldivclass}{o m}{%
\LWR@htmlelementclass{div}[#1]{#2}%
}
```

`\LWR@htmldivclassend` {⟨*class*⟩}

```
\newcommand*{\LWR@htmldivclassend}[1]{%
\LWR@htmlelementclassend{div}{#1}%
}
```

49.6 Single-line elements

A single-line element, without a paragraph tag for the line of text:

\LWR@htmlelementclassline {⟨*element*⟩} [⟨*style*⟩] {⟨*class*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
\LWR@stoppars
\LWR@subhtmlelementclass{#1}[#2]{#3}%
#4%
\LWR@htmltag{/#1}
\LWR@startpars
}
```

49.7 HTML5 semantic elements

\LWR@htmlelement {⟨*element*⟩}

```
\newcommand*{\LWR@htmlelement}[1]{%
\LWR@htmlblocktag{#1}
}
```

\LWR@htmlelementend {⟨*element*⟩}

```
\newcommand*{\LWR@htmlelementend}[1]{%
\LWR@stoppars
\LWR@htmltag{/#1}
\LWR@startpars
}

\end{warpHTML}
```

49.8 High-level block and inline classes

These are high-level commands which allow the creation of arbitrary block or inline sections which may be formatted with css.

Nullified versions are provided for print mode.

For other direct-formatting commands, see section 86.

Env BlockClass [⟨*style*⟩] {⟨*class*⟩} High-level interface for classes.

Ex: \begin{BlockClass}{class} text \end{BlockClass}

for PRINT output:

```
\begin{warpprint}
\NewDocumentEnvironment{BlockClass}{o m}{}{}%
```

```
\end{warpprint}
```

for HTML output:

```
\begin{warpHTML}
\NewDocumentEnvironment{LWR@print@BlockClass}{o m}{}{}%
\NewDocumentEnvironment{LWR@HTML@BlockClass}{o m}%
{%
    \LWR@origpar%
    \LWR@htmldivclass[#1]{#2}%
}
{\LWR@htmldivclassend{#2}}

\LWR@formattedenv{BlockClass}
\end{warpHTML}
```

`\BlockClassSingle` {⟨*class*⟩} {⟨*text*⟩} A single-line `<div>`, without a paragraph tag for the line of text.

for HTML & PRINT:

```
\begin{warpall}
\newcommand{\BlockClassSingle}[2]{#2}
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
\newcommand{\LWR@HTML@BlockClassSingle}[2]{%
\LWR@origpar%
\LWR@htmlelementclassline{div}{#1}{#2}%
}

\LWR@formatted{BlockClassSingle}
\end{warpHTML}
```

`\InlineClass` [⟨*style*⟩] {⟨*class*⟩} {⟨*text*⟩} High-level interface for inline span classes.

for PRINT output:

```
\begin{warpprint}
\NewDocumentCommand{\InlineClass}{o m +m}{#3}%
\end{warpprint}
```

for HTML output:

```
\begin{warpHTML}
\NewDocumentCommand{\LWR@print@InlineClass}{o m +m}{#3}%

\NewDocumentCommand{\LWR@HTML@InlineClass}{o m +m}{%
    \LWR@htmlspanclass[#1]{#2}{#3}%
}

\LWR@formatted{InlineClass}
\end{warpHTML}
```

Env `LWR@BlockClassWP` {⟨*WPstyle*⟩} {⟨*HTMLstyle*⟩} {⟨*class*⟩} Low-level interface for `<div>` classes with an automatic float ID. These are often used when `\ifbool{FormatWP}`.

for PRINT output:

```
\begin{warpprint}
\NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{}{}%
\end{warpprint}
```

for HTML output:

```
\begin{warpHTML}
\NewDocumentEnvironment{LWR@print@LWR@BlockClassWP}{m m m}{}{}%
\NewDocumentEnvironment{LWR@HTML@LWR@BlockClassWP}{m m m}%
{%
\LWR@stoppars%
\ifbool{FormatWP}%
{%
    \addtocounter{LWR@thisautoidWP}{1}%
    \LWR@htmltag{%
        div class="#3" %
        id="\LWR@print@mbox{autoidWP-\arabic{LWR@thisautoidWP}}"%
        \ifblank{#1}{}{ style="#1"}%
    }%
}% FormatWP
{% not FormatWP
    \LWR@htmltag{%
        div class="#3"%
        \ifblank{#2}{}{ style="#2"}%
    }%
}% not FormatWP
\LWR@startpars%
}
{\LWR@htmldivclassend{#3}}

\LWR@formattedenv{LWR@BlockClassWP}
\end{warpHTML}
```

49.9 Closing HTML tags

for HTML output:

```
\begin{warpHTML}
```

Sections H1, H2, etc. do not need a closing HTML tag, but we add a comment for readability:

```
\newcommand*{\LWR@printclosepart}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
\newcommand*{\LWR@printclosechapter}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
\newcommand*{\LWR@printclosesection}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
\newcommand*{\LWR@printclosesubsection}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
\newcommand*{\LWR@printclosesubsubsection}
```

```
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
\newcommand*{\LWR@printcloseparagraph}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
\newcommand*{\LWR@printclosesubparagraph}
    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```
\newcommand*{\LWR@printcloselistitem}
    {\LWR@htmltag{/li}}
\newcommand*{\LWR@printclosedescitem}
    {\LWR@htmltag{/dd}}
\newcommand*{\LWR@printcloseitemize}
    {\LWR@htmltag{/ul}}
\newcommand*{\LWR@printcloseenumerate}
    {\LWR@htmltag{/ol}}
\newcommand*{\LWR@printclosedescription}
    {\LWR@htmltag{/dl}}

\end{warpHTML}
```

50 Paragraph handling

These commands generate the HTML paragraph tags when allowed and required.

Paragraph tags are or are not allowed depending on many conditions. Section 51 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a LaTeX paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output:

```
\begin{warpHTML}
```

Ctr `LWR@spandepth` Do not create paragraph tags inside of an HTML span.

```
\newcounter{LWR@spandepth}
\setcounter{LWR@spandepth}{0}
```

Bool `LWR@doingstartpars` Tells whether paragraphs may be generated.

```
\newbool{LWR@doingstartpars}
\boolfalse{LWR@doingstartpars}
```

Bool LWR@doingapar Tells whether have actually generated and are currently processing paragraph text.

```
\newbool{LWR@doingapar}
\global\boolfalse{LWR@doingapar}
```

\LWR@ensuredoingapar If are about to print something visible, and if allowed to start a new paragraph, ensure that are LWR@doingapar, so that paragraph tags are placed:

```
\newcommand*{\LWR@ensuredoingapar}{%
\ifbool{LWR@doingstartpars}%
{\global\booltrue{LWR@doingapar}}%
{}%
}
```

\PN@parnotes@auto Redefined by **parnotes** to print paragraph notes at the end of each paragraph.

```
\def\PN@parnotes@auto{}%
```

\LWR@openparagraph

```
\newcommand*{\LWR@openparagraph}
{%
```

See if paragraph handling is enabled:

```
\ifbool{LWR@doingstartpars}%
{% handling pars
```

See if have already started a lateximage or a . If so, do not generate nested paragraph tags.

```
    \ifboolexpr{
        test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
        test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
    }% nested par tags?
```

If so: Do nothing if already started a lateximage page. Cannot nest a lateximage. Also do nothing if already inside a . Do not nest paragraph tags inside a .

```
    {}% no nested par tags
```

Else: No lateximage or has been started yet, so it's OK to generate paragraph tags.

```
    {% yes nest par tags
```

If **parnotes** is used, paragraph notes are inserted before starting the next paragraph:

```
        \PN@parnotes@auto%
```

The opening paragraph tag:

```
        \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Now have started a paragraph.

```
        \global\booltrue{LWR@doingapar}%
```

At the endof each paragraph, generate closing tag and do regular /par stuff. (Attempting to use the **everyhook** cr hook for \LWR@closeparagraph does not work well.)

```
        \let\par\LWR@closeparagraph%
    }% end of yes nest par tags
}% end of handling pars
{}% not handling pars
}
```

\LWR@closeparagraph

```
\newcommand*{\LWR@closeparagraph}
{%
% \LWR@traceinfo{LWR@closeparagraph}%
```

See if paragraph handling is enabled:

```
\ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
{% handling pars
```

See if already started a lateximage or a :

```
    \ifboolexpr{
        test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
        test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
    }%
```

Do nothing if already started a lateximage or a , but add a parbreak if in a span but not a lateximage.

```
    {% no nested par tags
```

```
        \ifboolexpr{
            test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}} and
            test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
        }%
        {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
        {}%
    }% no nested par tags
```

If have not already started a `lateximage` or a `<span>`:

```
    {% yes nest par tags
```

Print a closing tag and some extra vertical space.

```
        \unskip%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
```

No longer doing a paragraph:

```
        \global\boolfalse{LWR@doingapar}%
```

Disable the special `minipage` & `\hspace` interaction until a new minipage is found:

```
        \global\boolfalse{LWR@minipagethispar}%
```

If **parnotes** is used, paragraph notes are inserted after ending the previous paragraph:

```
        \PN@parnotes@auto%
    }% end of yes nest par tags
}% end of handling pars
```

Add a parbreak if in a span, but not in a table outside a row:

```
{% not handling pars
    \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
    {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
    {}%
}% not handling pars
```

In most cases, finish with a LaTeX `\par`, but in the case of paragraphs between lines in a tabular fetch the next token instead:

```
\ifboolexpr{%
    not bool {LWR@doingapar} and
    test {\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}} and
    test {
```

```
        \ifnumcomp{\value{LWR@tabulardepth}}{=}{\value{LWR@tabularpardepth}}
    } and
    bool {LWR@intabularmetadata} and
    not bool {LWR@tableparcell} and
    test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
}%
{%
    \LWR@getmynexttoken%
}{%
    \LWR@origpar%
}%
}

\end{warpHTML}
```

51 Paragraph start/stop handling

These commands allow/disallow the generation of HTML paragraph tags.

Section 50 has the commands which actually generate the tags.

The everyhook package is used to generate the opening paragraph tags. The closing tags are generated by \par.

for HTML output:

```
\begin{warpHTML}
```

\LWR@startpars

Begin handling HTML paragraphs. This allows an HTML paragraph to start, but one has not yet begun.

```
\newcommand*{\LWR@startpars}%
{%
% \LWR@traceinfo{LWR@startpars}%
```

Ignore if inside a span:

```
\ifnumcomp{\value{LWR@spandepth}}{>}{0}%
{}%
{%
```

See if currently handling HTML paragraphs:

```
    \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
    {}%
```

If not currently in paragraph mode:

```
    {%
```

At the start of each paragraph, generate an opening tag:

```
        \PushPreHook{par}{\LWR@openparagraph}%
```

At the end of each paragraph, generate closing tag and do regular /par actions:

```
        \let\par\LWR@closeparagraph

    }% an intentionally blank line
```

Are now handling paragraphs, but have not yet actually started one:

```
    \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
    \global\boolfalse{LWR@doingapar}%
}% nestspan
% \LWR@traceinfo{LWR@startpars: done}%
}
```

\LWR@stoppars Stop handling HTML paragraphs. Any currently open HTML paragraph is closed, and no more will be opened.

```
\newcommand*{\LWR@stoppars}%
{%
```

Ignore if inside a span:

```
\ifnumcomp{\value{LWR@spandepth}}{>}{0}%
{}%
{%
```

See if currently handling HTML paragraphs:

```
    \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
    {%
```

Print a closing tag:

```
        \unskip%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
        \global\boolfalse{LWR@doingapar}%
```

Disable the special `minipage` & `\hspace` interaction until a new minipage is found:

```
        \global\boolfalse{LWR@minipagethispar}

    }% an intentionally blank line
```

If was not in an HTML paragraph:

```
    {}%
```

See if currently allowing HTML paragraphs:

```
    \ifbool{LWR@doingstartpars}%
```

If so: clear the `par` hook to no longer catch paragraphs:

```
    {\ClearPreHook{par}}%
```

Else: Do nothing:

```
    {}%
```

No longer in paragraph mode:

```
    \global\setbool{LWR@doingstartpars}{false}%
```

No <p> tag to undo:

```
    \global\boolfalse{LWR@doingapar}%
}% nestspan
}

\end{warpHTML}
```

52 Page headers and footers

for HTML & PRINT:
```
\begin{warpall}
```

In the following, catcode is manually changed back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
\newcommand{\LWR@firstpagetop}{} % for the home page alone
\newcommand{\LWR@pagetop}{} % for all other pages
\newcommand{\LWR@pagebottom}{}
```

`\HTMLFirstPageTop` {⟨*text and logos*⟩}

```
\newcommand{\HTMLFirstPageTop}[1]{%
    \renewcommand{\LWR@firstpagetop}{#1}%
}
```

`\HTMLPageTop` {⟨*text and logos*⟩}

```
\newcommand{\HTMLPageTop}[1]{%
    \renewcommand{\LWR@pagetop}{#1}%
}
```

`\HTMLPageBottom` {⟨*text and logos*⟩}

```
\newcommand{\HTMLPageBottom}[1]{%
    \renewcommand{\LWR@pagebottom}{#1}%
}
```

```
\end{warpall}
```

53 CSS

for HTML output:
```
\begin{warpHTML}
```

`\LWR@currentcss` The css filename to use. This may be changed mid-document using `\CSSFilename`, allowing different css files to be used for different sections of the document.

```
\newcommand*{\LWR@currentcss}{lwarp.css}
```

`\CSSFilename` {⟨*new-css-filename.css*⟩} Assigns the css file to be used by the following HTML pages.

```
\newcommand*{\CSSFilename}[1]{%
\renewcommand*{\LWR@currentcss}{#1}%
\@onelevel@sanitize\LWR@currentcss%
}

\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\newcommand*{\CSSFilename}[1]{}
\end{warpprint}
```

54 Title, HTML meta author, HTML meta description

for HTML output:

```
\begin{warpHTML}
```

`\title` {⟨*title*⟩} Modified to remember `\thetitle`, which is used to set the HTML page titles.

```
\let\LWR@origtitle\title

\renewcommand*{\title}[1]{%
    \LWR@origtitle{#1}%
    \begingroup%
        \renewcommand{\thanks}[1]{}%
        \protected@xdef\thetitle{#1}%
    \endgroup%
}

\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

`\HTMLTitle` {⟨*Titlename*⟩} The Title to place into an HTML meta tag. The default is to use the document `\title`'s setting.

```
\providecommand{\thetitle}{}

\newcommand{\theHTMLTitle}{\thetitle}

\newcommand{\HTMLTitle}[1]{\renewcommand{\theHTMLTitle}{#1}}
```

`\HTMLAuthor` {⟨*authorname*⟩} The author to place into an HTML meta tag. If none given, the default is `\theauthor`, which is empty unless the **titling** package is used.

```
\providecommand{\theauthor}{}

\newcommand{\theHTMLAuthor}{\theauthor}

\newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
```

This is placed inside an HTML meta tag at the start of each file. This may be changed mid-document using `\HTMLDescription`, allowing different HTML descriptions to be used for different sections of the document.

⚠ Do not use double quotes, and do not exceed 150 characters.

`\HTMLDescription` {⟨*New HTML meta description.*⟩} Assigns the HTML file's description meta tag.

```
\newcommand{\LWR@currentHTMLDescription}{}

\newcommand{\HTMLDescription}[1]{%
\renewcommand{\LWR@currentHTMLDescription}{#1}
}

\end{warpall}
```

55 Footnotes

lwarp uses native LaTeX footnote code, although with its own `\box` to avoid the LaTeX output routine. The usual functions mostly work as-is.

The **footmisc** `stable` option is emulated by **lwarp**.

⚠ sectioning commands When using footnotes in sectioning commands, to generate consistent results between print and HTML, use the **footmisc** package with the `stable` option, provide a short TOC entry, and `\protect` the `\footnote`:

```
\usepackage[stable]{footmisc}
...
\subsection[Subsection Name]
    {Subsection Name\protect\footnote{A footnote.}}
```

If using **memoir** class, with which **lwarp** preloads **footmisc**, the `stable` option must be declared before **lwarp** is loaded:

```
\PassOptionsToPackage{stable}{footmisc}
\usepackage{lwarp}
...
```

Do not use a starred sectioning command. As an alternative, it may be possible to adjust `\secnumdepth` instead.

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handle differently.

55.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page or the section, depending on `FootnoteDepth`, using the LaTeX box `\LWR@footnotes`. Using this instead of the original `\footins` box avoids having footnotes be printed by the output routine, since footnotes should be printed per HTML page instead of per PDF page.

See section 55.4 for the implementation.

55.2 Minipage footnotes

See section 55.5 for how minipage footnotes are gathered. See section 85.3 for how minipage footnotes are placed into the document.

55.3 Titlepage thanks

See section 62.7 for titlepage footnotes.

55.4 Regular page footnote implementation

for HTML & PRINT:

```
\begin{warpall}
```

Ctr `FootnoteDepth` Default: 3

Determines how deeply to place footnotes in the HTML files, similar to `tocdepth`. The default of 3 places footnotes before each `\subsubsection` or higher. See table 7 for a table of LaTeX section headings.

```
\newcounter{FootnoteDepth}
\setcounter{FootnoteDepth}{3}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Patch LaTeX footnotes to use a new `\box` instead of an insert for **lwarp** footnotes. This avoids having the original `\footins` appear at the bottom of a `lateximage`, which is on its own new page.

```
\newbox\LWR@footnotes
```

Much of the following has unneeded print-mode formatting removed.

`\@makefntext` {⟨*text*⟩}

```
\long\def\@makefntext#1{\textsuperscript{\@thefnmark}~#1}
```

`\@makefnmark`

```
\def\@makefnmark{%
    \textsuperscript{\@thefnmark}%
}
```

Footnotes may be in regular text, in which case paragraphs are tagged, or in a table data cell or `lateximage`, in which case paragraph tags must be added manually.

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode `minipage`, but the footnotes are broken out by:

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext
```

`\LWR@footnotetext` {⟨*text*⟩}

```
\long\def\LWR@footnotetext#1{%
\LWR@traceinfo{LWR@footnotetext}%
\global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
    \unvbox\LWR@footnotes%
```

Remember the footnote number for `\ref`:

```
    \protected@edef\@currentlabel{%
        \csname p@footnote\endcsname\@thefnmark%
    }% @currentlabel
```

Open a group:

```
    \color@begingroup%
```

Use HTML superscripts in the footnote even inside a `lateximage`:

```
    \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
    \@makefntext{#1}%
```

Closing paragraph tag:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {\par}%
    {%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
    }%
```

Close the group:

```
    \color@endgroup%
}% vbox
```

Paragraph handling:

```
\LWR@ensuredoingapar%
}%
```

`\@footnotetext` {⟨*text*⟩}

```
\LetLtxMacro\@footnotetext\LWR@footnotetext
```

55.5 Minipage footnote implementation

Patch LaTeX minipage footnotes to use a new \box instead of an insert for **lwarp** minipage footnotes. This avoids having the original \@mpfootins appear at the bottom of a lateximage, which is on its own new page.

```
\newbox\LWR@mpfootnotes
```

\@mpfootnotetext {⟨*text*⟩}

```
\long\def\@mpfootnotetext#1{%
\LWR@traceinfo{@mpfootnotetext}%
\global\setbox\LWR@mpfootnotes\vbox{%
    \unvbox\LWR@mpfootnotes%
    \reset@font\footnotesize%
    \hsize\columnwidth%
    \@parboxrestore%
    \protected@edef\@currentlabel%
        {\csname p@mpfootnote\endcsname\@thefnmark}%
    \color@begingroup%
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%

    \@makefntext{%
        \ignorespaces#1%
    }%
```

Don't add the closing paragraph tag if are inside a lateximage:

```
    \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
    {}%
    {%
        \LWR@htmltagc{/\LWR@tagregularparagraph}%
        \LWR@orignewline%
    }%
    \color@endgroup%
}% vbox
```

Paragraph handling:

```
\LWR@ensuredoingapar%
\LWR@traceinfo{@mpfootnotetext: done}%
}
```

\thempfootnote Redefined to remove the \itshape, which caused an obscure compiling error in some situations.

```
\AtBeginDocument{
\def\thempfootnote{\@alph\c@mpfootnote}
}
```

55.6 Printing pending footnotes

\LWR@printpendingfootnotes Enclose the footnotes in a class, print, then clear.

```
\newcommand*{\LWR@printpendingfootnotes}{%
\ifvoid\LWR@footnotes\else
    \LWR@forcenewpage
    \begin{BlockClass}{footnotes}
    \LWR@origmedskip
    \unvbox\LWR@footnotes
    \setbox\LWR@footnotes=\vbox{}
    \end{BlockClass}
\fi
}
```

\LWR@maybeprintpendingfootnotes {⟨*depth*⟩} Used to print footnotes before sections only if formatting for an EPUB or word processor:

```
\newcommand*{\LWR@maybeprintpendingfootnotes}[1]{%
\ifboolexpr{
    not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
    bool{FormatEPUB} or
    bool{FormatWP}
}%
{\LWR@printpendingfootnotes}%
{}%
}
```

\LWR@printpendingmpfootnotes Enclose the minipage footnotes in a class, print, then clear.

```
\newcommand*{\LWR@printpendingmpfootnotes}{%
\ifvoid\LWR@mpfootnotes\else
    \LWR@forcenewpage
```

```
    \begin{BlockClass}{footnotes}
    \LWR@print@vspace*{\baselineskip}
    \unvbox\LWR@mpfootnotes
    \setbox\LWR@mpfootnotes=\vbox{}
    \end{BlockClass}
\fi
}

\end{warpHTML}
```

56 Marginpars

\marginpar [⟨*left*⟩] {⟨*right*⟩} \marginpar may contains paragraphs, but in order to remain inline with the surrounding text **lwarp** nullifies block-related macros inside the \marginpar. Paragraph breaks are converted to <br /> tags.

\marginparBlock [⟨*left*⟩] {⟨*right*⟩} To include block-related macros, use \marginparBlock, which takes the same arguments but creates a instead of a . A line break will occur in the text where the \marginBlock occurs.

for HTML output:

```
\begin{warpHTML}
```

\marginpar [⟨*left*⟩] {⟨*right*⟩}

```
\renewcommand{\marginpar}[2][]{%
\ifbool{FormatWP}%
{%
\begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
#2
\end{LWR@BlockClassWP}
}%
{%
    \LWR@htmlspanclass{marginpar}{#2}%
}%
}
```

\marginparBlock [⟨*left*⟩] {⟨*right*⟩}

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

HTML version.

```
\newcommand{\marginparBlock}[2][]{%
```

```
\ifbool{FormatWP}%
{%
\begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
#2
\end{LWR@BlockClassWP}
}%
{%
\begin{BlockClass}[width:2in; float:right; margin:10pt]{marginparblock}
#2
\end{BlockClass}
}%
}
```

\reversemarginpar

```
\renewcommand*{\reversemarginpar}{}
```

\normalmarginpar

```
\renewcommand*{\normalmarginpar}{}
```

```
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
```

\marginparBlock [⟨*left*⟩] {⟨*right*⟩}

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

Print version.

```
\LetLtxMacro\marginparBlock\marginpar
```

```
\end{warpprint}
```

57 Splitting HTML files

- Files are split according to `FileDepth` and `CombineHigherDepths`.
- Filenames are sanitized by `\LWR@filenamenoblanks`.
- `\LWR@newhtmlfile` finishes an HTML page, adds a comment to tell where and how to split the file, then starts a new HTML page.

for HTML & PRINT:

```
\begin{warpall}
```

Ctr `FileDepth` {⟨*section depth*⟩} determines how deeply to break into new HTML files, similar to `tocdepth`. The default of -5 produces one large HTML file.

```
\newcounter{FileDepth}
\setcounter{FileDepth}{-5}
```

Bool `CombineHigherDepths` Combile higher-level sections together into one file?

```
\newbool{CombineHigherDepths}
\booltrue{CombineHigherDepths}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

`\LWR@thisfilename` The currently-active filename or number.

```
\newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
\newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenamenoblanks` {⟨*filename*⟩}

Convert blanks into dashes, removes short words, store result in `\LWR@thisfilename`.

⚠ Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to `\protect` LaTeX commands which appear in section names and TOC captions.

```
\newcommand*{\LWR@filenamenoblanks}[1]{%
\begingroup
```

Locally temporarily disable direct-formatting commands, not used in filenames:

```
\LWR@nullfonts%
\renewcommand*{\LWR@htmltagc}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
\edef\LWR@thisnewfilename{#1}%
\LWR@traceinfo{LWR@filenamenoblanks edef: !\LWR@thisnewfilename!}%
\fullexpandarg%
```

Convert spaces into hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\LWR@origampersand}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textless}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textgreater}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\LWR@origpound}{-}[\LWR@thisnewfilename]

\StrSubstitute{\LWR@thisnewfilename}{\_}{-}[\LWR@thisnewfilename]

\StrSubstitute{\LWR@thisnewfilename}{\ }{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}%
    {\textbackslash}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{~{}}{-}[\LWR@thisnewfilename]
%     "~{}" for babel
\StrSubstitute{\LWR@thisnewfilename}{[}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{]}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{`}{-}[\LWR@thisnewfilename]
```

Convert short words:

```
\StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
```

```
\StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]
```

Convert multiple hyphens:

```
\StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
```

If pdfLaTeX and not `utf8` encoding, don't try to convert emdash, endash:

```
\ifPDFTeX%
\ifdefstring{\inputencodingname}{utf8}{%
\StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
%     emdash
\StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
%     endash
}{}%
\else% not PDFTeX
\StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
\StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
\fi%
```

Return the result:

```
\global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
\endgroup%
\LWR@traceinfo{LWR@filenamenoblanks: result is \LWR@thisfilename}%
}
```

Ctr `LWR@previousautopagelabel`

Remembers which autopage label was most recently generated. Used to avoid duplicates.

```
\newcounter{LWR@previousautopagelabel}
\setcounter{LWR@previousautopagelabel}{-1}
```

\LWR@newautopagelabel {⟨*pagenumber counter*⟩}

```
\newcommand*{\LWR@newautopagelabel}[1]{%
\ifnumequal{\value{LWR@previousautopagelabel}}{\value{page}}%
{}% no action if this autopage label has already been defined
{%
    \label{autopage-\arabic{#1}}%
    \setcounter{LWR@previousautopagelabel}{\value{page}}
}%
}
```

\LWR@customizedMathJax Additional MathJax definitions to be added to the start of each HTML page.

```
\newcommand*{\LWR@customizedMathJax}{}
```

\CustomizeMathJax MathJax does not have preexisting support every possible math function. Additional MathJax function definitions may be defined. These will be declared at the start of each HTML page, and thus will have a global effect.

Examples:

```
\CustomizeMathJax{
    \newcommand{\expval}[1]{\langle#1\rangle}
    \newcommand{\abs}[1]{\lvert#1\rvert}
}
\CustomizeMathJax{\newcommand{\arsinh}{\text{arsinh}}}
\CustomizeMathJax{\newcommand{\arcosh}{\text{arcosh}}}
\CustomizeMathJax{\newcommand{\NN}{\mathbb{N}}}
```

```
\newcommand*{\CustomizeMathJax}[1]{%
    \appto{\LWR@customizedMathJax}{%
        \(#1\)\par
    }%
}
```

\LWR@customizeMathJax

```
\newcommand{\LWR@customizeMathJax}{%
\ifbool{mathjax}{
\LWR@stoppars
\LWR@htmlcomment{Nullify \textbackslash{}ensuremath, footnotes for MathJax:}

\(\newcommand\ensuremath[1]{##1}\)

\(\newcommand\footnote[2][]{\text{( Footnote ##1 )}}\)

\(\newcommand\footnotemark[1][]{\text{( Footnote ##1 )}}\)
```

```
5299
5300 \LWR@htmlcomment{Additional customizations for MathJax:}
5301
5302 \LWR@customizedMathJax
5303
5304 \LWR@startpars
5305 }{}
5306 }

5307 \end{warpHTML}
```

for PRINT output:

```
5308 \begin{warpprint}
```

\CustomizeMathJax The print-mode version:

```
5309 \newcommand*{\CustomizeMathJax}[1]{}

5310 \end{warpprint}
```

for HTML output:

```
5311 \begin{warpHTML}
```

\LWR@newhtmlfile {⟨*section name*⟩}

Finishes the current HTML page with footnotes, footer, navigation, then starts a new HTML page with an HTML comment telling where to split the page and what the new filename and CSS are, then adds navigation, side TOC, header, and starts the text body.

```
5312 \newcommand*{\LWR@newhtmlfile}[1]{
5313 \LWR@traceinfo{LWR@newhtmlfile}
```

At the bottom of the ending file:

```
5314 \LWR@htmlelementclassend{section}{textbody}
5315
5316 \LWR@printpendingfootnotes
5317
```

No footer between files if EPUB:

```
5318 \ifbool{FormatEPUB}
5319 {}
5320 {
5321     \LWR@htmlelement{footer}
5322
5323     \LWR@pagebottom
```

```
5324
5325     \LWR@htmlelementend{footer}
5326 }
```

No bottom navigation if are finishing the home page or formatting for EPUB or a word-processor.

```
5327 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5328 {}
5329 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
5330 \LWR@stoppars
5331 \LWR@htmltag{/body}\LWR@orignewline
5332 \LWR@htmltag{/html}\LWR@orignewline
5333 \LWR@traceinfo{LWR@newhtmlfile: about to LWR@orignewpage}
5334 \LWR@orignewpage
5335
5336 \addtocounter{LWR@htmlfilenumber}{1}%
```

If using a filename, create a version without blanks. The filename without blanks will be placed into `\LWR@thisfilename`. If not using a filename, the file number will be used instead.

```
5337 \ifbool{FileSectionNames}%
5338 {\LWR@filenamenoblanks{#1}}
5339 {\renewcommand*{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}}
```

Include an HTML comment to instruct lwarpmk where to split the files apart. Uses pipe-separated fields for `split_html.gawk`. Uses monospaced font with ligatures disabled for everything except the title.

```
5340 \LWR@traceinfo{LWR@newhtmlfile: about to print start file}%
```

`\LWR@nullfonts` to allow math in a section name.

```
5341 \begingroup%
5342 \LWR@nullfonts%
5343 \LWR@htmlblockcomment{%
5344 |Start file|%
5345 \LWR@htmlsectionfilename{\LWR@thisfilename}|%
5346 }
5347 \endgroup%
```

At the top of the starting file:

```
\LWR@stoppars
```

If pdfLaTeX and not `utf8` encoding, use a hyphen instead of an emdash:

```
\ifPDFTeX%
\ifdefstring{\inputencodingname}{utf8}{%
\LWR@filestart{ -- #1}% there is an EMdash in front of the #1
}{
\LWR@filestart{ - #1}% hyphen
}
\else%
\LWR@filestart{ -- #1}% there is an EMdash in front of the #1
\fi%
```

Track the page numbers:

```
\setcounter{LWR@latestautopage}{\value{page}}%
\LWR@newautopagelabel{LWR@latestautopage}%
```

No navigation between files if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\LWR@topnavigation}
```

No header if between files if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{
    \LWR@htmlelement{header}

    \LWR@pagetop

    \LWR@htmlelementend{header}
}
```

Print title only if there is one. Skip if formatting for an EPUB or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\ifcsvoid{thetitle}{}{\LWR@printthetitle}}
```

No sidetoc if formatting for an epub or word processor:

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{\LWR@sidetoc}

```

Start of the <textbody>:

```
\LWR@htmlelementclass{section}{textbody}

```

Keep paragraph tags disabled for now:

```
\LWR@stoppars

```

If using MathJax, disable \ensuremath by printing a nullified definition at the start of each file, and add further customizations:

```
\LWR@customizeMathJax

\LWR@traceinfo{LWR@newhtmlfile: done}
}

\end{warpHTML}
```

58 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing LaTeX code and packages. Formatting is handled by css, so the emulated code has much less work to do than the print versions.

Unicode

Section names and the resulting filenames with accented characters are partially supported, depending on the ability of **pdflatex** to generate characters and **pdftotext** to read them. If extra symbols appear in the text, it may be that **pdflatex** is actually producing a symbol over or under a character, resulting in **pdftotext** picking up the accent symbol separately.

XeLaTeX and LuaLaTeX directly support accented section and file names.

for HTML output:

```
\begin{warpHTML}
```

58.1 User-level starred section commands

\ForceHTMLPage For HTML output, forces the next section to be on its own HTML page, if FileDepth allows, even if starred. For use with \printindex and others which generate a starred section which should be on its own HTML page. Also see \ForceHTMLTOC.

For print output, no effect.

```
\newbool{LWR@forcinghtmlpage}
\boolfalse{LWR@forcinghtmlpage}

\newcommand*{\ForceHTMLPage}{%
\global\booltrue{LWR@forcinghtmlpage}%
}
```

\ForceHTMLTOC For HTML output, forces the next section to have a TOC entry, even if starred. For use with \printindex and others which generate a starred section which should be in the TOC so that it may be accessed via HTML. Not necessary if used with **tocbibind**. Also see \ForceHTMLPage.

For print output, no effect.

```
\newbool{LWR@forcinghtmltoc}
\boolfalse{LWR@forcinghtmltoc}

\newcommand*{\ForceHTMLTOC}{%
\global\booltrue{LWR@forcinghtmltoc}%
}

\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\newcommand*{\ForceHTMLPage}{}
\newcommand*{\ForceHTMLTOC}{}
\end{warpprint}
```

for HTML output:

```
\begin{warpHTML}
```

58.2 Book class commands

\mainmatter ⚠ Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.

```
\newbool{LWR@mainmatter}
```

```
\DeclareDocumentCommand{\mainmatter}{}{%
\booltrue{LWR@mainmatter}%
}
```

`\frontmatter` Declare the front matter section of the document, using arabic numbering for the internal numbering. Does not reset the page number.

```
\DeclareDocumentCommand{\frontmatter}{}{%
\boolfalse{LWR@mainmatter}%
}
```

`\backmatter` Declare the back matter section of the document. Does not reset the page number.

```
\DeclareDocumentCommand{\backmatter}{}{%
\boolfalse{LWR@mainmatter}
}
```

58.3 Sectioning support macros

`\LWR@sectionumber` {⟨*section type*⟩}

Typeset a section number and its trailing space with CSS formatting:

```
\newcommand*{\LWR@sectionnumber}[1]{%
\InlineClass{sectionnumber}{#1}%
}
```

`autosec` A tag used by the TOC and index.

`\LWR@createautosec` {⟨*section type*⟩}

Create an autosection tag.

```
\newcommand*{\LWR@createautosec}[1]{%
\LWR@htmltag{#1 id="\LWR@print@mbox{autosec-\arabic{page}}"}%
}
```

`\LWR@pushoneclose` {⟨*depth*⟩} {⟨*printclose*⟩} Stacks the new sectioning level's closing tag, to be used when this section is closed some time later.

⚠ `\LWR@stoppars` must be executed first.

```
\NewDocumentCommand{\LWR@pushoneclose}{m m}{%
```

```
\LWR@traceinfo{LWR@pushoneclose #1}%
    \pushclose{#2}{#1}%
}
```

`\LWR@startnewdepth` {⟨*depth*⟩} {⟨*printclose*⟩}

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

⚠ `\LWR@stoppars` must be executed first.

```
\NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
\LWR@closeprevious{#1}%
```

Push a new section depth:

```
\LWR@pushoneclose{#1}{#2}%
}
```

Ctr `LWR@prevFileDepth` Remembers the previous `LWR@FileDepth`.

Initialized to a deep level so that any section will trigger a new HTML page after the home page.

```
\newcounter{LWR@prevFileDepth}
\setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\@seccntformat` {⟨*sectiontype*⟩}

```
\def\@seccntformat#1{\csname the#1\endcsname\protect\quad}
```

`\simplechapterdelim` Used by **tocbibind** and **anonchap**.

```
\newcommand*{\simplechapterdelim}{}
```

`\@chapcntformat` {⟨*sectiontype*⟩}

`\let` to `\@seccntformat` by default, but may be redefined by `\simplechapter` and `\restorechapter` from **tocbibind** or **anonchap**.

```
\let\@chapcntformat\@seccntformat
```

Ctr LWR@currentautosec

Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding `lateximage` will cause the page number to change by the time the following autosec label is created.

```
\newcounter{LWR@currentautosec}
```

\LWR@section * [⟨*TOC name*⟩] {⟨*name*⟩} {⟨*sectiontype*⟩}

The common actions for the high-level sectioning commands.

```
\DeclareDocumentCommand{\LWR@section}{m m m m}{%
\LWR@traceinfo{LWR@section |#2| |#3|}%
\LWR@traceinfo{LWR@section: not an empty section}%
\LWR@stoppars%
```

Cancel special `minipage` horizontal space interaction:

```
\global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for `\part*`, for **appendix**.

Generate a new LaTeX page so that TOC and index page number points to the section:

```
\LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
\IfBooleanT{#1}{\LWR@traceinfo{LWR@section: starred}}%
\ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
\ifthenelse{%
    \(%
        \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
        \(\cnttest{\@nameuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
        \(\boolean{LWR@forcinghtmlpage}\)%
    \)%
    \AND%
    \cnttest{\@nameuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
    \AND%
    \(%
        \NOT\boolean{CombineHigherDepths}\OR%
        \cnttest{\@nameuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
    \)%
    \AND%


    \(% phantomsection
        \NOT\isempty{#3}%
        \OR%
        \(\NOT\equal{#1}{\BooleanTrue}\)%
```

```
    \)%
}%
```

If so: start a new HTML file:

```
{% new file
    \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
    \IfNoValueTF{#2}%
```

If no optional entry

```
    {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
    {\LWR@newhtmlfile{#2}}%
}% new file
```

Else: No new HTML file:

```
{% not new file
```

Generate a new LaTeX page so that TOC and index page number points to the section:

```
    \LWR@traceinfo{LWR@section: not a new HTML file, about to LWR@orignewpage}%
    \LWR@orignewpage%

}% not new file
```

Remember this section's name for `\nameref`:

```
\IfValueT{#3}{%
\LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
\IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
}%
```

Print an opening comment with the level and the name; ex: "section" "Introduction" Footnotes may be used in section names, which would also appear in the HTML section opening comments, so the short TOC entry is used if possible, and a limited opening comment is made if the sectional unit is starred.

```

\ifbool{HTMLDebugComments}{%
    \begingroup%
```

```
    \LWR@nullfonts%
    \IfBooleanTF{#1}% starred
    {\LWR@htmlcomment{Opening #4*}}%
    {%
        \IfNoValueTF{#2}% short TOC
            {\LWR@htmlcomment{Opening #4 ``#3''}}%
            {\LWR@htmlcomment{Opening #4 ``#2''}}%
    }
    \endgroup%
}{}%

```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
\ifthenelse{%
    \cnttest{\@nameuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
}%
{\LWR@startpars}%
{}%
```

Create the opening tag with an autosec:

```
\LWR@traceinfo{LWR@section: about to LWR@createautosec}%
\LWR@createautosec{\@nameuse{LWR@tag#4}}%

\setcounter{LWR@currentautosec}{\value{page}}
```

Check if starred:

```
\IfBooleanTF{#1}%
{%
\LWR@traceinfo{LWR@section: starred}%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
\ifbool{LWR@forcinghtmltoc}%
{\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
{}%
}% starred
```

Not starred, so step counter and add to TOC:

```
{% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
    \ifthenelse{%
        \cnttest{\@nameuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
    }%
    {% if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For `article` class, **lwarp** assumes that all is mainmatter.

```
        \LWR@traceinfo{LWR@section: about to test main matter}%
        \ifbool{LWR@mainmatter}%
        {%
            \LWR@traceinfo{LWR@section: yes mainmatter}%
            \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
            \LWR@traceinfo{LWR@section: about to addcontentsline}%
            \addcontentsline{toc}{#4}%
            {%
                 \protect\numberline{\@nameuse{the#4}}%
                {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
            }%
            \LWR@traceinfo{LWR@section: finished addcontentsline}%
        }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
        {% not main matter
            \LWR@traceinfo{LWR@section: no main matter}%
            \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
        }% end of not main matter
    }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
    {%
        \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
    }%
```

For part, print the section type:

```
    \ifbool{LWR@mainmatter}%
    {%
        \ifthenelse{%
            \(\cnttest{\@nameuse{LWR@depth#4}}{<=}%
                {\value{secnumdepth}}\) \AND%
            \(\cnttest{\@nameuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)%
        }%
```

```
        {\@nameuse{#4name}~{}}%
        {}%
```

Print the section number:

```
        \LWR@traceinfo{LWR@section: about to print section number}%
        \ifthenelse{%
            \cnttest{\@nameuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
        }%
        {%
            \ifstrequal{#4}{chapter}%
            {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
            {\protect\LWR@sectionnumber{\@seccntformat{#4}}}%
        }%
        {}%
        \LWR@traceinfo{LWR@section: finished print section number}%
    }{}%
}% end of not starred
```

Print the section name:

```
\LWR@traceinfo{LWR@section: about to print the section name}%
#3%
```

Close the heading tag, such as /H2:

```
\LWR@traceinfo{LWR@section: about to close the heading tag}%
\LWR@htmltag{\@nameuse{LWR@tag#4end}}%
```

Generate a LaTeX label:

```
\LWR@traceinfo{LWR@section: about to create the LaTeX label}%
\LWR@newautopagelabel{LWR@currentautosec}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
\ifthenelse{%
    \cnttest{\@nameuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
}%
{\LWR@startpars}%
{}%
```

If not starred, remember the previous depth to possibly trigger a new HTML page.

HOWEVER, allow a `\part*` to start a new HTML page. This is used by **appendix**.

A starred section does not trigger a new HTML page at the beginning of this macro, so it should not affect it here at the end either. This became an issue when

a \listoftables was tested in the middle of the document. The \chapter* for the list was not allowing a new HTML page for the section following it while CombineHigherDepths was true.

```
\ifthenelse{%
    \NOT\equal{#1}{\BooleanTrue}\OR%
    \cnttest{\@nameuse{LWR@depth#4}}{=}{\LWR@depthpart}%
}%
{% not starred
    \setcounter{LWR@prevFileDepth}{\@nameuse{LWR@depth#4}}%
}% not starred
{}%
```

Reset to defaults if not a phantomsection:

```
\ifstrempty{#3}%
{}%
{%
\global\boolfalse{LWR@forcinghtmlpage}%
\global\boolfalse{LWR@forcinghtmltoc}%
}%
%
\LWR@traceinfo{LWR@section: done}%
}
```

58.4 \section and friends

\part * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\newcommand{\part@preamble}{}% for koma-script

\DeclareDocumentCommand{\part}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%

\LWR@section{#1}{#2}{#3}{part}%

\part@preamble% for koma-script
\renewcommand{\part@preamble}{}%
}
```

\chapter * [⟨*TOC name*⟩] [⟨*heading name*⟩] {⟨*name*⟩}

```
\let\@printcites\relax% for quotchap package

\newcommand{\chapter@preamble}{}% for koma-script

\@ifundefined{chapter}
{}
{%
\DeclareDocumentCommand{\chapter}{s o o m}{%
\IfValueTF{#2}{
\LWR@traceinfo{chapter #2}%
}{
\LWR@traceinfo{chapter #4}%
}
\LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%

\LWR@section{#1}{#2}{#4}{chapter}%

\@printcites% for quotchap package

\chapter@preamble% for koma-script
\renewcommand{\chapter@preamble}{}%
}
}
```

`\section` * [⟨*TOC name*⟩] [⟨*heading name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\section}{s o o m}{%
\IfValueTF{#2}{
\LWR@traceinfo{section #2}%
}{
\LWR@traceinfo{section #4}%
}
\LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
\LWR@stoppars%

\LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%

\LWR@section{#1}{#2}{#4}{section}%
}
```

`\subsection` * [⟨*TOC name*⟩] {⟨*name*⟩}

```
\DeclareDocumentCommand{\subsection}{s o m}{%
\LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
\LWR@stoppars%
```

```
5644
5645 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
5646
5647 \LWR@section{#1}{#2}{#3}{subsection}%
5648 }
```

\subsubsection * [⟨*TOC name*⟩] {⟨*name*⟩}

```
5649 \DeclareDocumentCommand{\subsubsection}{s o m}{%
5650 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
5651 \LWR@stoppars%
5652
5653 \LWR@startnewdepth{\LWR@depthsubsubsection}%
5654 {\LWR@printclosesubsubsection}%
5655
5656 \LWR@section{#1}{#2}{#3}{subsubsection}%
5657 }
```

\paragraph * [⟨*TOC name*⟩] {⟨*name*⟩}

```
5658 \DeclareDocumentCommand{\paragraph}{s o m}{%
5659 \LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
5660 \LWR@stoppars%
5661
5662 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
5663
5664 \LWR@section{#1}{#2}{#3}{paragraph}%
5665 }
```

\subparagraph * [⟨*TOC name*⟩] {⟨*name*⟩}

```
5666 \DeclareDocumentCommand{\subparagraph}{s o m}{%
5667 \LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
5668 \LWR@stoppars%
5669
5670 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
5671
5672 \LWR@section{#1}{#2}{#3}{subparagraph}%
5673 }

5674 \end{warpHTML}
```

59 Starting a new file

for HTML & PRINT:

```
5675 \begin{warpall}
```

\HTMLLanguage Default language for the HTML lang tag.

```
\newcommand*{\LWR@currentHTMLLanguage}{en-US}

\newcommand*{\HTMLLanguage}[1]{%
\renewcommand*{\LWR@currentHTMLLanguage}{#1}%
}

\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

\LWR@filestart {⟨*title_suffix*⟩}

Creates the opening HTML tags.

```
\newcommand*{\LWR@filestart}[1]{
\LWR@traceinfo{LWR@filestart !#1!}
```

Locally temporarily disable direct-formatting commands:

```
\begingroup
\LWR@nullfonts
```

Create the page's HTML header:

```
\LWR@htmltag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
\LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
\LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
\LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
\ifthenelse{\equal{\theHTMLAuthor}{}}%
{}%
{\LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline}%
```

lwarp is the generator:

```
\LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
    \LWR@orignewline
```

If there is a description, add it now:

```
\ifdefempty{\LWR@currentHTMLDescription}{}{%
\LWR@htmltag{%
meta name="description" content="\LWR@currentHTMLDescription" /}%
    \LWR@orignewline
}%
```

Mobile-friendly viewport:

```
\LWR@htmltag{meta name="viewport" %
content="width=device-width, initial-scale=1.0" /}%
    \LWR@orignewline
```

IE patch:

```
\LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
\LWR@htmltag{%
script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"}%
\LWR@htmltag{/script}\LWR@orignewline
\LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
\ifthenelse{\equal{\theHTMLTitle}{}}%
{}%
{\LWR@htmltag{title}\theHTMLTitle#1\LWR@htmltag{/title}\LWR@orignewline}%
```

The page's stylesheet:

```
\LWR@htmltag{%
link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
\LWR@orignewline
```

Optional MathJax support. The HTML tags must be turned off during the verbatim input, and the paragraph handling which was turned on at the end of verbatim input must be immediately turned off again.

```
\ifbool{mathjax}%
{%
    \begingroup%
    \LWR@restoreoriglists%
    \boolfalse{LWR@verbtags}
```

```
    \verbatiminput{lwarp_mathjax.txt}%
    \booltrue{LWR@verbtags}
    \endgroup%
    \LWR@stoppars
}% end of mathjax
{}%
```

End of the header:

```
\LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
\LWR@htmltag{body}\LWR@orignewline
\endgroup
\LWR@traceinfo{LWR@filestart: done}
}

\end{warpHTML}
```

60 Starting HTML output

for HTML output:

```
\begin{warpHTML}
```

`\LWR@LwarpStart` Executed at the beginning of the entire document.

```
\catcode`\$=\active
\newcommand*{\LWR@LwarpStart}
{%
\LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
\ifbool{FormatWP}{%
    \setcounter{FileDepth}{-5}%
    \boolfalse{HTMLDebugComments}%
}{}
```

Expand and detokenize `\HomeHTMLFilename` and `\HTMLFilename`:

```
\edef\LWR@strresult{\HomeHTMLFilename}
\edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
\edef\LWR@strresult{\HTMLFilename}
\edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
\LWR@origonecolumn%
\LWR@origpagestyle{empty}%
```

No black box for overfull lines:

```
\overfullrule=0pt
```

Reduce chance of line overflow in verbatim environments:

```
\LWR@print@scriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
\LWR@print@raggedright%
\LetLtxMacro{\\}{\LWR@endofline}%
```

Spread the lines for **pdftotext** to read them well:

```
\linespread{1.3}%
```

For **pdftotext** to reliably identify paragraph splits:

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
\immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space around the caption in the HTML:

```
\setlength{\belowcaptionskip}{0ex}
\setlength{\abovecaptionskip}{0ex}
```

Redefine the plain page style to be empty when used by index pages:

```
\renewcommand{\ps@plain}{}
```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```
\LetLtxMacro\LWR@origtabular\tabular
\LetLtxMacro\LWR@origendtabular\endtabular
```

```
\LetLtxMacro\tabular\LWR@tabular
\LetLtxMacro\endtabular\endLWR@tabular
```

Float captions:

```
\let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in **amsmath** environments and is also patched by **cleveref**.

Label in HTML

```
\let\LWR@origltx@label\ltx@label
\let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
\let\TeX\LWR@TeX
\let\LaTeX\LWR@LaTeX
\let\LuaTeX\LWR@LuaTeX
\let\LuaLaTeX\LWR@LuaLaTeX
\let\XeTeX\LWR@XeTeX
\let\XeLaTeX\LWR@XeLaTeX
\let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
\global\boolfalse{LWR@doingapar}
\global\boolfalse{LWR@doingstartpars}
```

Document and page settings:

```
\mainmatter
\LWR@origpagenumbering{arabic}
```

Start a new HTML file and a header:

```
\LWR@traceinfo{LWR@lwarpStart: Starting new file.}
\LWR@filestart{}
\LWR@traceinfo{LWR@lwarpStart: Generating first header.}
\LWR@htmltag{header}\LWR@orignewline
\LWR@startpars
\LWR@firstpagetop
\LWR@stoppars
\LWR@htmltag{/header}\LWR@orignewline
\LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
\LWR@htmltag{section class="textbody"}
```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native LaTeX environments, as well as those provided by **enumitem**, **enumerate**, and **paralist**.

```
\LWR@patchlists
```

Ensure that math mode is active to call **lwarp**'s patches:

```
\catcode`\$=\active
```

Required for `\nameref` to work with SVG math:

```
\immediate\write\@mainaux{\catcode`\string$\active}%
\LetLtxMacro\LWR@syntaxhighlightone$% balance for editor syntax highlighting
```

Allow HTML paragraphs to begin:

```
\LWR@startpars
```

If using MathJax, disable `\ensuremath` by printing a nullified definition at the start of each file, and add further customizations:

```
\LWR@customizeMathJax

\LWR@traceinfo{LWR@lwarpStart: done}
}
\catcode`\$=3% math shift until lwarp starts

\end{warpHTML}
```

61 Ending HTML output

for HTML output:

```
\begin{warpHTML}
```

`\LWR@requesttoc` {⟨*boolean*⟩} {⟨*suffix*⟩} Requests that a toc, lof, or lot be generated.

```
\newcommand*{\LWR@requesttoc}[2]{%
\ifbool{#1}
{
    \expandafter\newwrite\@nameuse{tf@#2}
    \immediate\openout \@nameuse{tf@#2} \jobname.#2\relax
}{}
}
```

\LWR@LwarpEnd Final stop of all HTML output:

```
\newcommand*{\LWR@LwarpEnd}
{
\LWR@stoppars
\LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
\LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
\LWR@printpendingfootnotes
```

Create the footer:

```
\LWR@htmlelement{footer}

\LWR@pagebottom

\LWR@htmlelementend{footer}
```

No bottom navigation if are finishing the home page, or if formatting for an EPUB or word processor.

Presumably has a table-of-contents.

```
\ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
{}
{
    \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
}

\LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
\LWR@htmltag{/body}\LWR@orignewline
\LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
\LWR@orignewpage
```

For lateximage commands:

```
\immediate\closeout\LWR@lateximagesfile
}

\end{warpHTML}
```

62 Title page

package support
⚠ load order

lwarp supports the native LaTeX titling commands, and also supports the packages **authblk** and **titling**. If both are used, **authblk** should be loaded before **titling**.

\published and \subtitle

If using the **titling** package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 62.8.

affiliation

lwarp provides for the \author macro an additional \affiliation macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using **titlingpage**'s \theauthor in the main text.

reusing titlepage information

The **titling** package maintains the definitions of \thetitle, \theauthor, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the \thanks and \affiliation removed, and for \author the \and is replaced to generate a simple inline list of authors separated by commas. Note: \theauthor does not work well with **authblk** unless the traditional LaTeX syntax is used.

⚠ \theauthor, **authblk**

custom titlepages

\printtitle, \printauthor, etc., are provided for use inside a custom titlepage or titlingpage environment, and these retain the \thanks and \affiliation.

\printthanks

\printthanks has been added to force the printing of thanks inside a titlingpage environment when \maketitle is not used.

⚠

Inside a \titlepage or \titlingpage environment, use \thanks instead of \footnote for acknowledgements, etc.

62.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

\author {⟨*author*⟩} While using \maketitle and print mode, the author is treated as a single-column tabular and the \and feature finishes the current tabular then starts a
\and

new one for the next author. Each author thus is placed into its own `tabular`, and an affiliation may be placed on its own line such as

```
\author{Name \\ Affiliation \and Second Name \\ Second Affiliation}
```

For HTML, the entire author block is placed inside a `<div>` of class `author`, and each individual author is inside a `<div>` of class `oneauthor`.

`\@title` `\@author` `\@date`

`\@title`, `\@author`, etc. store the values as originally assigned, including any `\thanks`, `\and`, or `\affiliation`. These are low-level macros intended to be used by other macros only inside a `titlepage` or `titlingpage`, and are used by `\maketitle`. The author is printed inside a single-column `tabular`, which becomes multiple single-column `tabulars` if multiples authors are included. For HTML these `tabulars` become side-by-side `<div>`s of class `oneauthor`, all of which are combined into one `<div>` of class `author`.

`\printtitle` `\printauthor` `\printdate`

`\printtitle`, etc. are user-level macros intended to be used in custom `titlepage` or `titlingpage` environments in cases where `\maketitle` is not desired. These commands preserve the `\thanks`, etc., and should not be used in the main text.

`\thetitle` `\theauthor` `\thedate`

`\thetitle`, `\theauthor`, and `\thedate` are available if **titling** has been loaded, and are sanitized user-level versions from which have been removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

`\HTMLPageBottom` `{`⟨*text*⟩`}`

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

⚠ `\theauthor`

`\theauthor` does not work well if **authblk** is used. If `\theauthor` is important, it is recommended to use the standard LaTeX syntax for `\author`, optionally with **lwarp**'s `\affiliation` macro as well.

⚠ affiliations

After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

`\affiliation`

A solution, provide here, is to define a macro `\affiliation` which, during `\maketitle`, starts a new row and adds the affiliation, but after `\maketitle` is finished `\affiliation`

is re-defined to discard its argument, thus printing only the author names when \author is later used inline.

62.2 \if@titlepage

for HTML & PRINT:

```
\begin{warpall}
```

\if@titlepage

Some classes do not provide \if@titlepage. In this case, provide it and force it false.

```
\ifcsvoid{@titlepagefalse}{
    \newif\if@titlepage
    \@titlepagefalse
}{}

\end{warpall}
```

62.3 Changes for \affiliation

\affiliation {⟨*text*⟩}

Adds the affiliation to the author for use in \maketitle.

Inside titlepage, this macro prints its argument. Outside, it is null.

for HTML & PRINT:

```
\begin{warpall}
\providerobustcmd{\affiliation}[1]{}
\end{warpall}
```

for PRINT output:

```
\begin{warpprint}

\AtBeginEnvironment{titlepage}{
\renewrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
}

\AtBeginDocument{
\@ifpackageloaded{titling}{
\AtBeginEnvironment{titlingpage}{
\renewrobustcmd{\affiliation}[1]{\\ \textsc{\small#1}}
}
}{}% titling loaded
}% AtBeginDocument

\end{warpprint}
```

for HTML output:
```
\begin{warpHTML}
```

Env titlepage Sets up a of class titlepage. Provided even for **memoir** class, since it is used by \maketitle.

```
\DeclareDocumentEnvironment{titlepage}{}
{
\renewrobustcmd{\affiliation}[1]{\\ \InlineClass{affiliation}{##1}}
\LWR@printpendingfootnotes
\LWR@forcenewpage
\BlockClass{titlepage}
}
{
\endBlockClass
\LWR@printpendingfootnotes
}
```

```
\end{warpHTML}
```

62.4 Printing the thanks

for HTML & PRINT:
```
\begin{warpall}
```

\printthanks Forces the \thanks to be printed.

This is necessary in a titlingpage environment when \maketitle was not used.

```
\newcommand*{\printthanks}{\@thanks}
```

```
\end{warpall}
```

62.5 Printing the title, etc. in HTML

The following are for printing the title, etc. in a titlepage or a titlingpage in HTML:

for HTML output:
```
\begin{warpHTML}
```

\printtitle

```
\newcommand*{\printtitle}
{
\LWR@stoppars
```

```
\LWR@htmltag{\LWR@tagtitle}%
\@title%
\LWR@htmltag{\LWR@tagtitleend}
\LWR@startpars
}
```

\LWR@printthetitle A private version which prints the title without footnotes, used to title each HTML page.

```
\newcommand*{\LWR@printthetitle}
{
\LWR@stoppars
\LWR@htmltag{\LWR@tagtitle}%
\thetitle%
\LWR@htmltag{\LWR@tagtitleend}
\LWR@startpars
}
```

\printauthor HTML version.

```
\newcommand*{\printauthor}{
```

The entire author block is contained in a named author:

```
\begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
\renewcommand{\and}{%
\end{BlockClass}
\begin{BlockClass}{oneauthor}
}
```

Individual authors are contained in a named oneauthor:

```
\begin{BlockClass}{oneauthor}
\@author
\end{BlockClass}
\end{BlockClass}
}
```

\printdate

```
\newcommand*{\printdate}{%
\begin{BlockClass}{titledate}
\@date
```

```
\end{BlockClass}
}

\end{warpHTML}
```

62.6 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

for PRINT output:

```
\begin{warpprint}
```

`\printtitle`

```
\newcommand*{\printtitle}{{\Huge\@title}}
```

`\printauthor` Print mode.

```
\newcommand*{\printauthor}
    {{\large\begin{tabular}[t]{c}\@author\end{tabular}}}
```

`\printdate`

```
\newcommand*{\printdate}{{\small\textit{\@date}}}

\end{warpprint}
```

62.7 `\maketitle` for HTML output

An HTML `<div>` of class `titlepage` is used.

`\thanks` are a form of footnotes used in the title page. See section 55 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output:

```
\begin{warpHTML}

\@ifclassloaded{memoir}
{
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@arabic\c@footnote}%
```

```
}
}{% not memoir
\if@titlepage
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@arabic\c@footnote}%
}
\else
\newcommand{\LWR@setfootnoteseries}{%
    \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
}
\fi
}% not memoir
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
\newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
\LWR@setfootnoteseries%
\def\@makefnmark{\textsuperscript{\thefootnote}}
```

`\thefootnote` ⇒ `\nameuse`{arabic}{footnote}, or
`\thefootnote` ⇒ `\nameuse`{fnsymbol}{footnote}

Redefine the footnote text:

```
\long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
\textsuperscript{\@thefnmark}~%
```

`\makethanksmark` ⇒ `\thanksfootmark` ⇒ `\tamark` ⇒
`\@thefnmark` ⇒ `\itshape` a (or similar)

Print the text:

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

`\@fnsymbol` {⟨*counter*⟩}

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used \| which was not being seen by **pdftotext**.

```
\def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
   \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLunicode{2016}}\or
   **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
   \HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerr\fi}
```

\maketitle HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the **titling** package is adapted, simplified, and modified for HTML output.

```
\renewcommand*{\maketitle}{%
```

An HTML titlepage is used for all classes.

```
\begin{titlepage}
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
```

Close the HTML titlepage div and cleanup:

```
\end{titlepage}
\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
}
```

\@maketitle HTML mode. Typesets the title, etc.:

```
\DeclareDocumentCommand{\@maketitle}{}{%
    \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
    \@title
    \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
    \begin{BlockClass}{author}
```

For **IEEEtran** class:

```
    \renewcommand*{\cr}{}
    \renewcommand*{\crcr}{}
    \renewcommand*{\noalign}{}

        \renewcommand{\and}{
            \end{BlockClass}
            \begin{BlockClass}{oneauthor}
        }
        \begin{BlockClass}{oneauthor}
            \@author
        \end{BlockClass}
    \end{BlockClass}
    \begin{BlockClass}{titledate}
    \@date
    \end{BlockClass}
}
```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
\newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
\@thanks
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
}
```

```
\end{warpHTML}
```

62.8 \published and \subtitle

\subtitle and \published

To add \subtitle and \published to the titlepage, load the **titling** package and use \AddSubtitlePublished in the preamble.

The default lwarp.css has definitions for the published and subtitle classes.

If **titling** is loaded, \AddSubtitlePublished creates a number of additional macros, and also assigns some of the **titling** hooks. If **titling** is not loaded, \AddSubtitlePublished creates null macros.

⚠ **titling** hooks

Do not use \AddSubtitlePublished if the user has patched the **titling** hooks for some other reason. Portions are marked \warpprintonly to reduce extra tags in HTML. Similarly, BlockClass has no effect in print mode. Thus, the following may be marked warpall.

for HTML & PRINT:

```
\begin{warpall}
```

\AddSubtitlePublished

Adds \published and \subtitle, and related.

```
\newcommand*{\AddSubtitlePublished}{%
\@ifpackageloaded{titling}{% yes titling package
    \newcommand{\@published}{}%
    \newcommand{\published}[1]{\gdef\@published{##1}}%
    \renewcommand*{\maketitlehooka}{\printpublished}%
    \newcommand*{\printpublished}{%
        \warpprintonly{\begin{center}\unskip}%
        \begin{BlockClass}{published}%
        \warpprintonly{\large\itshape}%
        \@published%
        \end{BlockClass}%
        \warpprintonly{\end{center}}%
    }%
    \newcommand{\@subtitle}{}%
    \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}%
    \renewcommand*{\maketitlehookb}{\printsubtitle}%
    \newcommand*{\printsubtitle}{%
        \warpprintonly{\begin{center}\unskip}%
        \begin{BlockClass}{subtitle}%
        \warpprintonly{\Large\itshape}%
        \@subtitle%
        \end{BlockClass}%
        \warpprintonly{\end{center}}%
    }%
```

```
}% yes titling package
{% no titling package
    \newcommand{\published}[1]{}%
    \newcommand*{\printpublished}{}%
    \newcommand{\subtitle}{}%
    \newcommand*{\printsubtitle}%
}% no titling package
}% \AddSubtitlePublished

\end{warpall}
```

63 Abstract

The following code replaces the LaTeX default, and will itself be replaced later if the **abstract** package is loaded.

for HTML output:

```
\begin{warpHTML}
```

`\abstractname` User-redefinable title for the abstract.

Also over-written by the **babel** package.

```
\providecommand*{\abstractname}{Abstract}
```

Some classes allow an optional name, so it is allowed here.

Env `abstract`

```
\DeclareDocumentEnvironment{abstract}{O{\abstractname}}
{
\LWR@forcenewpage
\BlockClass{abstract}
\BlockClassSingle{abstracttitle}{#1}
}
{
\endBlockClass
}

\end{warpHTML}
```

64 Quote and verse

64.1 Attributions

\attribution For use with quote, quotation, verse:

Ex: `"A quotation." \attribution{\textsc{Author Name}\\\textsl{Book Title}}`

for HTML output:

```
\begin{warpHTML}
\newcommand{\attribution}[1]{%
    \begin{BlockClass}{attribution}
    #1
    \end{BlockClass}
}
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
\newcommand{\attribution}[1]{
    \begin{flushright}
    \unskip
    #1
    \end{flushright}%
}
\end{warpprint}
```

64.2 Quotes, quotations

for HTML output:

```
\begin{warpHTML}
```

Env quote

```
\renewenvironment*{quote}
{
\LWR@forcenewpage
\LWR@htmlblocktag{blockquote}
}
{\LWR@htmlblocktag{/blockquote}}

\renewenvironment*{quotation}
{
\LWR@forcenewpage
\LWR@htmlblocktag{blockquotation}
}
{\LWR@htmlblocktag{/blockquotation}}
```

```
\end{warpHTML}
```

64.3 Verse

When using **verse** or **memoir**, always place a \\ after each line.

\attrib The documentation for the **verse** and **memoir** packages suggest defining an \attrib command, which may already exist in current documents, but it will only work for print output. **lwarp** provides \attribution, which works for both print and HTML output. To combine the two so that \attrib is used for print and \attribution is used for HTML:

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

Len \vleftskip
Len \vleftmargini
Len \HTMLvleftskip
Len \HTMLleftmargini

These lengths are used by **verse** and **memoir** to control the left margin, and they may already be set by the user for print output. New lengths \HTMLvleftskip and \HTMLleftmargini are provided to control the margins in HTML output. These new lengths may be set by the user before any verse environment, and persist until they are manually changed again. One reason to change \HTMLleftmargini is if there is a wide \flagverse in use, such as the word "Chorus", in which case the value of \HTMLleftmargini should be set to a wide enough length to contain "Chorus". The default is wide enough for a stanza number.

⚠ spacing Horizontal spacing relies on **pdftotext**'s ability to discern the layout (-layout option) of the text in the HTML-tagged PDF output. For some settings of \HTMLleftmargini or \HTMLleftskip the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

64.3.1 LaTeX core verse environment

for HTML output:

```
\begin{warpHTML}
```

Env verse

```
\renewenvironment{verse}
               {\let\\\newline% lwarp
                \list{}{\itemsep      \z@
                        \itemindent   -1.5em%
                        \listparindent\itemindent
```

```
                        \rightmargin  \leftmargin
                        \advance\leftmargin 1.5em}%
                 \item\relax}
               {\endlist}

\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

64.3.2 verse and memoir

The following lengths are used by **verse** and **memoir**. They may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

Len `\HTMLvleftskip` Sets `\vleftskip` inside a `verse` environment in HTML.

```
\newlength{\HTMLvleftskip}
\setlength{\HTMLvleftskip}{1em}
```

Len `\HTMLleftmargini` Sets `\leftmargini` inside a `verse` environment in HTML.

```
\newlength{\HTMLleftmargini}
\setlength{\HTMLleftmargini}{4.5em}

\end{warpall}
```

65 Verbatim and tabbing

for HTML & PRINT:

```
\begin{warpall}
```

Len `\VerbatimHTMLWidth` Width to use in HTML `Verbatim` environment.

This width is used when placing line numbers to the right. Ignored during print output.

```
\newlength{\VerbatimHTMLWidth}
\setlength{\VerbatimHTMLWidth}{4in}
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML

head.

```
\newbool{LWR@verbtags}
\booltrue{LWR@verbtags}
```

\LWR@atbeginverbatim [⟨*1: style*⟩] {⟨*2: negative* \baselineskip *\vspace*⟩} {⟨*3: class*⟩}

Encloses a verbatim environment with the given CSS class.

```
\newcommand*{\LWR@atbeginverbatim}[3][]
{%
```

Avoid excessive space between lines:

```
\setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
\LWR@stoppars%
```

Create a new `pre` of the given class. The tags may temporarily be turned off for internal use, such as loading the MathJax script.

```
\ifbool{LWR@verbtags}{%
    \LWR@htmltag{pre class="#3"
        \ifthenelse{\equal{#1}{}}{}{style="#1"}%
    }%
    \LWR@orignewline% pre
    \leavevmode\unskip\LWR@print@vspace*{-#2\baselineskip}%
}{}%
```

Use a mono-spaced font to preserve horizontal positioning. If horizontal alignment is important for the user, use a mono-spaced font in the CSS for the `verse` class.

```
\begingroup%
```

```
% \LWR@print@normalsize%
\LWR@origttfamily%
\LWR@print@small%
```

Since inside a <pre>, restore the original list processing:

```
\LWR@restoreoriglists%
```

Turn off **babel-french** extra space before punctuation:

```
\LWR@FBcancel%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain LaTeX \hspace functionality:

```
\LWR@select@print@hspace%
}
```

\LWR@afterendverbatim {⟨*negative* *baselineskip* *vspace*⟩}

Finishes enclosing a verbatim environment.

```
\newcommand*{\LWR@afterendverbatim}[1]{%
\endgroup%
\par%
```

At the end of the environment, close the pre:

```
\ifbool{LWR@verbtags}{%
    \LWR@print@vspace*{-#1\baselineskip}%
    \noindent\LWR@htmltag{/pre}\LWR@orignewline% pre
}{}%
```

Resume regular paragraph handling:

```
\LWR@startpars%
}
```

\verbatiminput {⟨*filename*⟩}

Patch \verbatiminput to add HTML tags:

```
\let\LWRV@origverbatim@input\verbatim@input

\renewcommand{\verbatim@input}[2]{%
\ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
\LWR@atbeginverbatim{2.5}{Verbatim}%
\LWRV@origverbatim@input{#1}{#2}%
\LWR@afterendverbatim{1.5}%
}
```

Env verbatim

```
\AfterEndPreamble{
\LWR@traceinfo{Patching verbatim.}
\AtBeginEnvironment{verbatim}{%
\LWR@forcenewpage%
\LWR@atbeginverbatim{2.5}{verbatim}%
}
```

```
\AfterEndEnvironment{verbatim}{%
    \LWR@afterendverbatim{1}%
}
}
```

Env `tabbing` The `tabbing` environment works, except that SVG math and `lateximages` do not yet work inside the environment.

math in `tabbing` If math is used inside `tabbing`, place `tabbing` inside a `lateximage` environment, which will render the entire environment as a single SVG image.

```
\newcommand*{\LWR@HTML@tabbing}{%
\LWR@forcenewpage%
\LWR@atbeginverbatim{3}{tabbing}%
\LWR@print@tabbing%
}

\newcommand*{\LWR@HTML@endtabbing}{%
    \LWR@print@endtabbing%
    \LWR@afterendverbatim{1}%
}

\LWR@formatted{tabbing}
\LWR@formatted{endtabbing}


\end{warpHTML}
```

66 Theorems

`\newtheorem` {⟨*text*⟩} [⟨*counter*⟩] -or- [⟨*oldname*⟩] {⟨*text*⟩}

A few minor changes are made to supply HTML tags.

- The entire theorem is placed into a `<div>` of class `theoremcontents`.
- The label for each theorem is placed inside a `<span>` of class `theoremlabel`.
- The contents are placed inside a `<div>` of class theoremcontents.

for HTML output:

```
\begin{warpHTML}
```

`\@begintheorem` {⟨*name*⟩} {⟨*number*⟩}

```
\renewcommand{\@begintheorem}[2]{%
\LWR@forcenewpage
\BlockClass{theoremcontents}
\trivlist
\item[\InlineClass{theoremlabel}{#1\ #2\ }]\itshape
}
```

\@opargbegintheorem {⟨*name*⟩} {⟨*number*⟩} {⟨*oparg*⟩}

```
\renewcommand{\@opargbegintheorem}[3]{%
\LWR@forcenewpage
\BlockClass{theoremcontents}
\trivlist
\item[\InlineClass{theoremlabel}{#1\ #2\ (#3)\ }]\itshape
}
```

\@endtheorem

```
\renewcommand*{\@endtheorem}{%
\endtrivlist
\endBlockClass% theoremcontents
}
```

```
\end{warpHTML}
```

67 Lists

The environments `itemize`, `enumerate`, and `description` are patched when **lwarp** is started. These patches support the standard LaTeX environments, as well as those of **enumerate**, **enumitem**, and **paralist**, and at least the French version of **babel**. Additional patches are done on a package-specific basis.

The LaTeX source for `itemize` and `enumerate` are found in `source2e`, but the source for `description` is found in `article.cls`, etc.

empty item

To have an empty item, use `\mbox{}` or a trailing backslash. This forces a new line in print output, matching the new line which will appear in HTML output. Ex:

```
begin{itemize}
item \mbox{}
    \begin{itemize}
...
    \end{itemize}
```

```
item \
    \begin{itemize}
  ...
    \end{itemize}
```

\makelabel While inside a list environment, **lwarp** nullifies a number of TEX horizontal skip and fill commands, allowing the user to define \makelabel for print mode while HTML mode ignores those commands.

⚠ label font When defining \makelabel in a list environment, use \textbf etc. instead of \bfseries.

67.1 List environment

for HTML output:
```
\begin{warpHTML}
```

\LWR@printcloselist May be locally redefined by enumerate or description.

```
\newcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
```

\LWR@printopenlist May be locally redefined by enumerate or description.

```
\newcommand*{\LWR@printopenlist}{ul style="\LWR@print@mbox{list-style-type:none}"}
```

\@mklab Removes PDF spacing.

```
\AtBeginDocument{
\def\@mklab#1{%
%      \hfil %
    #1}
\let\makelabel\@mklab
}
```

\@donoparitem Modified for HTML output by replacing TEX boxes with plain text. Also removes PDF spacing.

```
\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 Modified for HTML output by replacing TEX boxes with plain text. Also removes PDF spacing.

```
\def\LWR@HTML@item[#1]{%
\LWR@traceinfo{@item}
  \if@noparitem
    \@donoparitem
  \else
%     \if@inlabel
%       \indent
        \par
%     \fi
    \ifhmode
%       \unskip\unskip
        \par
    \fi
    \if@newlist
      \if@nobreak
        \@nbitem
      \else
%         \addpenalty\@beginparpenalty
%         \addvspace\@topsep
%         \addvspace{-\parskip}%
      \fi
    \else
%       \addpenalty\@itempenalty
%       \addvspace\itemsep
    \fi
    \global\@inlabeltrue
  \fi
%   \everypar{%
    \@minipagefalse
    \global\@newlistfalse


%     \if@inlabel
%       \global\@inlabelfalse


%       {\setbox\z@\lastbox
%        \ifvoid\z@
%          \kern-\itemindent
%        \fi}%
```

```
%       \box\@labels
%       \penalty\z@
%     \fi

%     \if@nobreak
%       \@nobreakfalse
%       \clubpenalty \@M
%     \else
%       \clubpenalty \@clubpenalty
%       \everypar{}%
%     \fi}%

  \if@noitemarg
    \@noitemargfalse
    \if@nmbrlist

      \refstepcounter\@listctr
    \fi
  \fi

    \makelabel{#1} % extra space
%   \sbox\@tempboxa{\makelabel{#1}%
%   \global\setbox\@labels\hbox{%
%     \unhbox\@labels
%     \hskip \itemindent
%     \hskip -\labelwidth
%     \hskip -\labelsep
%     \ifdim \wd\@tempboxa >\labelwidth
%       \box\@tempboxa

%     \else
%       \hbox to\labelwidth {\unhbox\@tempboxa}%
%     \fi
%     \hskip \labelsep}%
  \ignorespaces%
}
```

`\@nbitem`

```
\def\@nbitem{%
%   \@tempskipa\@outerparskip
%   \advance\@tempskipa -\parskip
%   \addvspace\@tempskipa
}
```

`\LWR@listitem` [⟨*label*⟩]

Handles \item inside a list, itemize, or enumerate.

See \LWR@openparagraph where extra \hspace is used to leave room for the label while inside a list during paragraph construction.

```
\newcommand*{\LWR@listitem}{%
\LWR@stoppars%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
\LWR@htmltag{li}%
\LWR@startpars%
\LWR@origitem%
}
```

\LWR@nulllistfills Nullifies various TEX fill commands, in case they are used inside \makelabel. Problems are caused when these are nullified all the time.

```
\newcommand*{\LWR@nulllistfills}{%
\renewcommand*{\hss}{}%
\renewcommand*{\llap}[1]{##1}%
\renewcommand*{\rlap}[1]{##1}%
\renewcommand*{\hfil}{}%
\renewcommand*{\hfilneg}{}%
\renewcommand*{\hfill}{}%
}
```

Env list {⟨*label*⟩} {⟨*commands*⟩}

```
\newcommand*{\LWR@liststart}{%
\LWR@traceinfo{LWR@liststart}%
\LWR@stoppars%
\LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
\LWR@htmltag{\LWR@printopenlist}\LWR@orignewline%
\LWR@startpars%
\setlength{\topsep}{0pt}%
\setlength{\partopsep}{0pt}%
\setlength{\itemsep}{0pt}%
\setlength{\parsep}{0pt}%
\setlength{\leftmargin}{0pt}%
\setlength{\rightmargin}{0pt}%
\setlength{\listparindent}{0pt}%
\setlength{\itemindent}{0pt}%
\setlength{\labelsep}{1em}%
\LWR@nulllistfills%
}
```

```
\newcommand*{\LWR@listend}{%
\LWR@traceinfo{LWR@listend}%
```

```
\LWR@stoppars%
\LWR@closeprevious{\LWR@depthlist}%
\LWR@startpars%
}
```

67.2 Itemize

`\LWR@itemizeitem` [⟨*label*⟩]

Handles `\item` inside an itemize or enumerate.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
\newcommand*{\LWR@itemizeitem}{%
\LWR@stoppars%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
\LWR@htmltag{li}%
\LWR@startpars%
\LWR@origitem%
}
```

Env `itemize` [⟨*options*⟩]

```
\newcommand*{\LWR@itemizestart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\renewcommand*{\LWR@printopenlist}{ul style="\LWR@print@mbox{list-style-type:none}"}
\let\item\LWR@itemizeitem%
\LWR@nulllistfills%
}
```

67.3 Enumerate

An HTML unordered list is used with customized LaTeX-generated labels.

Env `enumerate` [⟨*options*⟩]

```
\newcommand*{\LWR@enumeratestart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
\renewcommand*{\LWR@printopenlist}{ul style="\LWR@print@mbox{list-style-type:none}"}
\let\item\LWR@itemizeitem%
\LWR@nulllistfills%
}
```

67.4 Description

\LWR@descitem [⟨*label*⟩] Handles an \item inside a description.

```
\newcommand*{\LWR@descitem}[1][]%
{%
\LWR@stoppars%
\LWR@setlatestname{#1}%
\LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem}%
```

Temporarily disable \hspace, which article.cls, etc. use per \item for descriptions only. This causes **lwarp** to mistakenly place an empty span between HTML list tags.

```
\LWR@select@html@nohspace%
```

Process the original \item code:

```
\LWR@origitem[]%
```

Restore \hspace for use in the item text:

```
\LWR@select@html@hspace%

\LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
\LWR@orignewline%
\LWR@htmltag{dd}%
\LWR@startpars%
}
```

Env description [⟨*options*⟩]

```
\newcommand*{\LWR@descriptionstart}{%
\renewcommand*{\LWR@printcloselist}{\LWR@printclosedescription}
\renewcommand*{\LWR@printopenlist}{dl}
\let\item\LWR@descitem%
\LWR@nulllistfills%
}
```

67.5 Patching the lists

\LWR@patchlists Patches list environments.

\LWR@patchlists remembers \item as defined by whatever packages have been loaded, then patches the itemize, enumerate, and description environments and

\item. This works with the native LaTeX environments, as well as those provided by **enumitem**, **enumerate**, and **paralist**.

```
\newcommand*{\LWR@patchlists}{%
    \LetLtxMacro\item\LWR@listitem%
    \LetLtxMacro\@item\LWR@HTML@item%
    \renewcommand*{\@trivlist}{%
        \LWR@traceinfo{@trivlist start}%
        \LWR@liststart%
        \LWR@orig@trivlist%
        \LWR@traceinfo{@trivlist done}%
    }%
    \renewcommand*{\trivlist}{%
        \LWR@traceinfo{trivlist}%
        \LWR@origtrivlist%
    }%
    \renewcommand*{\endtrivlist}{%
        \LWR@traceinfo{endtrivlist start}%
        \LWR@origendtrivlist\LWR@listend%
        \LWR@traceinfo{endtrivlist done}%
    }%
    \renewcommand*{\itemize}{%
        \LWR@itemizestart\LWR@origitemize%
    }%
    \renewcommand*{\enumerate}{%
        \LWR@enumeratestart\LWR@origenumerate%
    }%
    \renewcommand*{\description}{%
        \LWR@descriptionstart\LWR@origdescription%
    }%
}
```

\LWR@restoreoriglists Restores the original trivlist environment.

```
\newcommand*{\LWR@restoreoriglists}{%
\LWR@traceinfo{LWR@restoreoriglists}%
\LetLtxMacro\item\LWR@origitem%
\LetLtxMacro\@item\LWR@orig@item%
\let\@trivlist\LWR@orig@trivlist%
\let\trivlist\LWR@origtrivlist%
\let\endtrivlist\LWR@origendtrivlist%
\LetLtxMacro\itemize\LWR@origitemize%
\LetLtxMacro\enditemize\LWR@endorigitemize%
\LetLtxMacro\enumerate\LWR@origenumerate%
\LetLtxMacro\endenumerate\LWR@endorigenumerate%
\LetLtxMacro\description\LWR@origdescription%
\LetLtxMacro\enddescription\LWR@endorigdescription%
\let\@mklab\LWR@orig@mklab%
\let\makelabel\LWR@origmakelabel%
```

```
\let\@donoparitem\LWR@orig@donoparitem%
\let\@nbitem\LWR@orig@nbitem%
}

\end{warpHTML}
```

68 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax of the LaTeX core and the various tabular-related packages.

68.1 Limitations

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, * column specifiers, **siunitx** S columns, or the packages **multirow**, **longtable**, **supertabular**, or **xtab**.

Defining environments:

⚠ misplaced alignment alignment tab character &

- When defining environments or macros which include `tabular` and instances of the `&` character, it may be necessary to make `&` active before the environment or macro is defined, then restore `&` to its default catcode after, using the following commands. These are are ignored in print mode.

```
\StartDefiningTabulars
<define macros or environments using tabular and &
here>
\StopDefiningTabulars
```

⚠ floatrow

This includes before and after defining any macro which used `\ttabbox` from **floatrow**.

⚠ `tabular` inside another environment

- When creating a new environment which contains a `tabular` environment, **lwarp**'s emulation of the tabular does not automatically resume when the containing enviroment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\StopDefiningTabulars
```

Cell contents:

⚠ paragraphs

- Multiple paragraphs in one cell of a `p`, `b`, `m` column must have `\newline` between paragraphs.

⚠ `\multirow`

- For **multirow**, insert `\mrowcell` into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

```
... &  \multirow{2}{.5in}{text}  & ...
... &  \mrowcell                 & ...
```

`vposn` Note that recent versions of **multirow** include a new optional `vposn` argument.

- The **multirow** documentation regarding colored cells recommends using a negative number of rows. This will not work with **lwarp**, so `\warpprintonly` and `\warpHTMLonly` must be used to make versions for print and HTML.
- See section 285.2 for `\multicolumnrow`.

⚠ `\multicolumn` & `\multirow`

lwarp does not support directly combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for `\multicolumn` come first, followed by the five arguments for `\multirow`, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines, `\mcolrowcell` must be used for each cell of each column and each row to be skipped:

```
... &  \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}  & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
... &  \mcolrowcell          &          \mcolrowcell   & ...
```

vposn

Note that recent versions of **multirow** include a new optional `vposn` argument.

⚠ macro in a table
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \\
```

Column specifiers:

⚠ * column specification

- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of `@` and `!` is used at each column, and they are used in that order.

\multirow

- In `\multirow` cells, the print version may have extra instances of `<`, `>`, `@`, and `!` cells on the second and later rows in the `\multirow` which do not appear in the HTML version.

⚠ \newcolumntype

- `\newcolumntype` is ignored; unknown column types are set to `l`.

Rules:

- Doubled `\hlines`, `\midrules`, and vertical rules are supported.

vertical rules

- Vertical rules next to either side of an `@` or `!` column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with `@` or `!` columns, and full-width rules ignore trim.

full-width rules

- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use `\cmidrule` followed by `\bottomrule`, it may be necessary to use:

```
\cmidrule{2-3} \\[-2ex]
\bottomrule
```

The optional `-2ex` is ignored in HTML but improves the visual formatting in the print output.

⚠ \warpprintonly
misplaced \noalign

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warpHTML` environment, if a "misplaced `\noalign`" error occurs, change

```
This & That \endhead
```

to

```
\warpprintonly{This & That \endhead}
```

and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevent to HTML output.

colortbl:

⚠ row/cell color

Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

colortbl ignores the overhang arguments.

Other:

- **tabularx** ignores the width, but `X` columns do produce paragraph columns or multicolumns.

`longtable` headings

- For **longtable**, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.

⚠ S columns

- For S columns (from the **siunitx** package), while producing print output, anything non-numeric must be placed inside { } braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by **lwarp**'s tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

for HTML output:

```
\begin{warpHTML}
```

68.2 Temporary package-related macros

These macros are temporary placeholders for macros defined by various packages. If the relevent package is not loaded, these placeholders are used instead.

68.2.1 arydshln

Emualated by the original LaTeX non-dashed versions.

```
\LetLtxMacro\hdashline\hline
\LetLtxMacro\cdashline\cline
\LetLtxMacro\firsthdashline\hline
\LetLtxMacro\lasthdashline\hline
```

68.3 Token lookahead

Used by `\LWR@futurenonspacelet` to look at the next token.

\LWR@mynexttoken

```
\newcommand\LWR@mynexttoken\relax
```

\futurelet copies the next token then executes a function to analyze

\LWR@futurenonspacelet does the same, but ignores intervening white space

Based on the **booktabs** style:

\LWR@futurenonspacelet

```
\def\LWR@futurenonspacelet#1{\def\LWR@cs{#1}%
\afterassignment\LWR@fnslone\let\nexttoken= }
\def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
\def\LWR@fnsltwo{%
\expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
\else\let\next=\nexttoken\fi\next}
\def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
\newcommand*{\LWR@getmynexttoken}{%
\LWR@traceinfo{LWR@getmynexttoken}%
% nothing must follow this next line
\LWR@futurenonspacelet\LWR@mynexttoken\LWR@tabledatacolumntag
}
```

68.4 Tabular variables

Bool LWR@startedrow True if should print a row tag before this column.

```
\newbool{LWR@startedrow}
\boolfalse{LWR@startedrow}
```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```
\newbool{LWR@tabularcelladded}
\boolfalse{LWR@tabularcelladded}
```

Ctr LWR@hlines Number of \hlines or \midrules above the next row.

```
\newcounter{LWR@hlines}
```

Ctr `LWR@hdashedlines` Number of **arydshln** dashed lines above the next row.

```
\newcounter{LWR@hdashedlines}
```

Bool `LWR@doingtbrule` True if the next row will have a top/bottom rule above it.

```
\newbool{LWR@doingtbrule}
\boolfalse{LWR@doingtbrule}
```

Bool `LWR@doingcmidrule` True if the next row will have a cmidrule above it.

This is used by `\LWR@tabularfinishrow` to force a final empty row to create the border for the `\cmidrule`.

```
\newbool{LWR@doingcmidrule}
\boolfalse{LWR@doingcmidrule}
```

Bool `LWR@tableparcell` True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
\newbool{LWR@tableparcell}
```

Bool `LWR@skippingmrowcell` True if are doing an empty `\multirow` cell, and thus there is no data tag to close.

```
\newbool{LWR@skippingmrowcell}
```

Bool `LWR@skippingmcolrowcell` True if are doing an empty `\multicolumnrow` cell, and thus there is no data tag to close, and do not print `@` and `!` columns.

```
\newbool{LWR@skippingmcolrowcell}
```

Bool `LWR@skipatbang` True if just finished a `\multicolumn` so should not create the trailing `@` or `!` columns table data cells.

```
\newbool{LWR@skipatbang}
```

Bool `LWR@emptyatbang` True if finishing a row and should print empty `@` or `!` column table data cells.

```
\newbool{LWR@emptyatbang}
```

Bool `LWR@intabularmetadata` True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
\newbool{LWR@intabularmetadata}
\boolfalse{LWR@intabularmetadata}
```

Ctr `LWR@tabularDepth` Tracks whether `&` is being used inside a `tabular`.

```
\newcounter{LWR@tabulardepth}
\setcounter{LWR@tabulardepth}{0}
```

Ctr `LWR@tabularpardepth` Tracks whether should look ahead at the next token when encountering a `\par` while processing tabular contents.

When `LWR@tabularpardepth` is deeper than `LWR@tabulardepth` then **lwarp** has started looking at the contents of the `tabular`, and thus any `\par`s encountered must be followed by another token lookahead.

```
\newcounter{LWR@tabularpardepth}
\setcounter{LWR@tabularpardepth}{0}
```

```
\newcommand*{\LWR@colsresult}{}%temp storage for column format results
\newcommand*{\LWR@pposition}{}
\newcommand*{\LWR@pleft}{}
\newcommand*{\LWR@pright}{}
```

`LWR@tablecolspec` Holds the parsed column specification, of total width `LWR@tabletotalLaTeXcols`, not counting `@` and `!` columns.

Will contain a string such as `llrrccpc`, exactly one letter per LaTeX table column, without `@`, `!`, `>`, `<`, or the vertical bar.

`\LWR@strresult` Holds the result of `Str` functions.

```
\providecommand*{\LWR@strresult}{}
\providecommand*{\LWR@strresulttwo}{}
```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
\newcommand*{\LWR@origcolspec}{}
```

Ctr `LWR@tablecolspecwidth` Holds the number of tokens in the table columns specification.

This is includes one for each `@`, `!`, `<`, `>` column, and also one for each of the parameters of `p`, `@`, `!`, `<`, `>` columns, and three for each `D` column.

(This is not the total # of LaTeX columns in the table.)

```
\newcounter{LWR@tablecolspecwidth}
```

Ctr `LWR@tablecolspecindex` While parsing the LaTeX table column specification, starts at 1 and is incremented per token of the specification.

```
\newcounter{LWR@tablecolspecindex}
```

Ctr LWR@tableLaTeXcolindex — While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
\newcounter{LWR@tableLaTeXcolindex}
```

Ctr LWR@tabletotalLaTeXcols — While parsing a table column specification, begins at 0 and increments by 1 per LaTeX table column. Eventually holds the final number of LaTeX table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in LWR@tabletotalLaTeXcols.)

```
\newcounter{LWR@tabletotalLaTeXcols}
```

Ctr LWR@tabletotalLaTeXcolsnext — Holds the next LaTeX table column index while parsing, equal to one more than LWR@tabletotalLaTeXcols.

```
\newcounter{LWR@tabletotalLaTeXcolsnext}
```

LWR@colatspec — A data array of specifications for @ columns. The leftmost's index is leftedge, the others are counter values. See section 39.

LWR@colbangspec — A data array of specifications for ! columns. The leftmost's index is leftedge, the others are counter values. See section 39.

LWR@colbeforespec — A data array of specifications for > columns.

LWR@colafterspec — A data array of specifications for < columns.

LWR@colbarspec — A data array of specifications for vertical rules.

68.5 Handling &, @, !, and bar

For technical discussion regarding problems redefining \&, See:
http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860

\LWR@instertatbangcols

```
\newcommand*{\LWR@insertatbangcols}{%
\ifbool{LWR@skipatbang}%
{}%
{%
    \LWR@printatbang{at}{\arabic{LWR@tableLaTeXcolindex}}%
    \LWR@printatbang{bang}{\arabic{LWR@tableLaTeXcolindex}}%
}%
}
```

\LWR@closetabledatacell If LWR@skippingmrowcell or LWR@skippingmcolrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
\newcommand*{\LWR@closetabledatacell}{%
\global\booltrue{LWR@intabularmetadata}%
\ifbool{LWR@exitingtabular}{}%
{% not exiting tabular
    \ifboolexpr{bool{LWR@skippingmrowcell} or bool{LWR@skippingmcolrowcell}}%
    {%
```

If not skipping a \multicolumnrow cell, insert the @ and ! columns after this non-existant column.

```
        \ifbool{LWR@skippingmcolrowcell}%
            {}%
            {\LWR@insertatbangcols}%
    }%
    {% not skippingmrowcell
```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```
        \unskip%
        \ifboolexpr{%
            bool{LWR@tabularmutemods} or
            bool{LWR@skipatbang} or
            bool{LWR@emptyatbang}
        }%
        {}%
        {\LWR@getexparray{LWR@colafterspec}{\arabic{LWR@tableLaTeXcolindex}}}%
```

Close paragraphs:

```
        \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
        \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell.

Close any color s.

```
        \whileboolexpr{test {\ifnumcomp{\value{LWR@cellcolordepth}}{>}{0}}}{%
            \LWR@htmltag{/div}\LWR@orignewline%
            \addtocounter{LWR@cellcolordepth}{-1}%
        }%
```

Skip the @ and ! cells if are closing a multicolumn cell.

```
        \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
```

```
        \global\booltrue{LWR@tabularcelladded}%
        \LWR@insertatbangcols%
    }% not skipping mrowcell
}% not exiting tabular
\global\boolfalse{LWR@skippingmrowcell}%
\global\boolfalse{LWR@skippingmcolrowcell}%
\global\boolfalse{LWR@skipatbang}%
```

Color control. Column is set by >{} for each cell, so it must be cleared here.

```
\gdef\LWR@cellHTMLcolor{}
\gdef\LWR@columnHTMLcolor{}
\setcounter{LWR@cellcolordepth}{0}
}
```

When not used inside a `tabular`, `&` performs its original function as recorded here (with catcode 4).

```
\let\LWR@origampmacro&
```

```
\end{warpHTML}
```

68.5.1 Handling &

for HTML output:

```
\begin{warpHTML}
```

`&` Will behave depending on whether it is being used inside `tabular`.

`&` is redefined to test whether it is inside a tabular environment, in which case it performs special processing for HTML conversion. If not, it behaves normally.

```
\newcommand*{\LWR@tabularampersand}{%
\LWR@traceinfo{LWR@tabularampersand}%
\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}%
{%
```

If not skipping a multirow cell, close the current data cell.

```
    \unskip%
    \LWR@closetabledatacell%
```

Move to the next column.

```
    \addtocounter{LWR@tableLaTeXcolindex}{1}%
```

Have not yet added data in this column:

```
    \global\boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
    \LWR@getmynexttoken%
}%
```

If not inside a tabular, performs the original action:

```
{\LWR@origampmacro}%
}
```

& is left with its original catcode for now.

tikz package seems to require & be left alone until after **tikz** has been loaded. Also, **cleveref** uses the ampersand in one of its options.

& is made active inside a `tabular`.

& is left alone when in math alignments.

68.5.2 Filling an unfinished row

`\LWR@tabularfinishrow` Adds empty table cells if necessary to finish the row.

At the end of the table, if any bottom rules are requested then an empty row must be generated to form the borders which show the rules.

```
\newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
\ifboolexpr{%
    not bool {LWR@exitingtabular} or%
    bool{LWR@doingtbrule} or%
    bool{LWR@doingcmidrule} or%
    test{\ifnumcomp{\value{LWR@hlines}}{>}{0}} or%
    test{\ifnumcomp{\value{LWR@hdashedlines}}{>}{0}} or%
    bool{LWR@startedrow}%
}{%
```

To temporarily turn off `LWR@exitingtabular` so that table data tags will still be generated:

If generating a final row for the \bottomrule borders, turn off the @, !, <, and > column output:

```
\ifbool{LWR@exitingtabular}{%
    \global\booltrue{LWR@tabularmutemods}%
}{%
    \global\boolfalse{LWR@tabularmutemods}%
}%
```

Locally reenable the table data tags until finished with the final row:

```
\global\boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
\whileboolexpr{%
    test {
        \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}
    } or %
    (%
        bool{LWR@intabularmetadata} and%
        not bool{LWR@tabularcelladded} and%
        test {
            \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{=}{\value{LWR@tabletotalLaTeXcols}}
        }%
    )%
}%
{%
    \LWR@tabledatasinglecolumntag%
```

The following is essentially \LWR@tabularampersand with LWR@emptyatbang added to empty the following cells:

```
    \LWR@closetabledatacell%
    \addtocounter{LWR@tableLaTeXcolindex}{1}%
    \global\boolfalse{LWR@tabularcelladded}%
    \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
    \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}%
        {\LWR@getmynexttoken}%
        {}%
}%
```

Reenable the original LWR@exitingtabular to close the entire table:

```
\ifbool{LWR@tabularmutemods}{%
```

```
    \global\booltrue{LWR@exitingtabular}%
}{%
    \global\boolfalse{LWR@exitingtabular}%
}%
\global\boolfalse{LWR@tabularmutemods}%

\global\boolfalse{LWR@emptyatbang}%
}{}% ifboolexpr
}
```

68.6 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

\LWR@tabularendofline

```
\NewDocumentCommand{\LWR@tabularendofline}{s o}{%
```

Finish the row:

```
\ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}%
    {\LWR@tabularfinishrow}%
    {\LWR@closetabledatacell}%
\LWR@htmltag{/tr}\LWR@orignewline%
```

xcolor row color support:

```
\@rowc@lors%
```

No longer inside a data cell:

```
\global\booltrue{LWR@intabularmetadata}%
```

Not yet started a table row:

```
\global\boolfalse{LWR@startedrow}%
```

Additional setup:

```
\setcounter{LWR@hlines}{0}%
\setcounter{LWR@hdashedlines}{0}%
\global\boolfalse{LWR@doingtbrule}%
\global\boolfalse{LWR@doingcmidrule}%
```

```
\LWR@clearmidrules%
\gdef\LWR@rowHTMLcolor{}%
```

Start at first column:

```
\setcounter{LWR@tableLaTeXcolindex}{1}%
```

Have not yet added data in this column:

```
\global\boolfalse{LWR@tabularcelladded}%
```

Allow TEX to flush the pending paragraph. Not doing so causes a slowdown for very large tables.

```
\LWR@stoppars
\LWR@origpar
```

Look at the next token to decide between single column data tag or a special case:

```
\LWR@getmynexttoken%
}
```

68.7 Looking ahead in the column specifications

`\LWR@columnspeclookahead` `{`⟨*offset*⟩`}`

Looks `offset` tokens ahead in the column specification, setting `\LWR@strresulttwo`.

The `w` column alignment will be seen as a single unit such as `{c}`.

```
\newcommand*{\LWR@columnspeclookahead}[1]{%
\setcounter{LWR@tempcountone}{\value{LWR@tablecolspecindex}}%
\addtocounter{LWR@tempcountone}{#1}%
\fullexpandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tempcountone}}[\LWR@strresulttwo]%
```

Get the contents of the first group in `\LWR@strresulttwo`:

```
\exploregroups%
\StrChar{\LWR@strresulttwo}{1}[\LWR@strresulttwo]%
\noexploregroups%
}
```

68.8 Parsing @, >, <, !, bar columns

Holds the parsed argument for `@`, `>`, `<`, or `!` columns:

```
\newcommand*{\LWR@colparameter}{}
```

`\LWR@parseatcolumn` Handles `@{text}` columns.

```
\newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
\LWR@traceinfo{at column}%
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
\LWR@traceinfo{about to read the next token:}%
\expandarg%
\StrChar{\LWR@origcolspec}%
    {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
\fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
\LWR@traceinfo{have now read the next token}%
\ifnumcomp{\value{LWR@tabletotalLaTeXcols}}{=}{0}%
{% left edge of the table:
    \LWR@traceinfo{at the left edge}%
    \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
    \LWR@traceinfo{at the left edge: %
    \LWR@getexparray{LWR@colatspec}{leftedge}}%
}%
{% not at the left edge:
    \LWR@traceinfo{not at the left edge}%
    \LWR@setexparray{LWR@colatspec}%
        {\arabic{LWR@tabletotalLaTeXcols}}{\LWR@colparameter}%
    \LWR@traceinfo{at \arabic{LWR@tabletotalLaTeXcols}: %
    \LWR@getexparray{LWR@colatspec}{\arabic{LWR@tabletotalLaTeXcols}}}%
}%
\let\LWR@colparameter\relax%
\global\booltrue{LWR@validtablecol}%
}
```

`\LWR@parsebangcolumn` Handles `!{text}` columns.

```
\newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
\LWR@traceinfo{bang column}%
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
\LWR@traceinfo{about to read the next token:}%
\expandarg%
\StrChar{\LWR@origcolspec}%
    {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
\fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
\LWR@traceinfo{have now read the next token}%
\ifnumcomp{\value{LWR@tabletotalLaTeXcols}}{=}{0}%
{% left edge of the table:
    \LWR@traceinfo{at the left edge}%
    \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
}%
{% not at the left edge:
    \LWR@traceinfo{not at the left edge}%
    \LWR@setexparray{LWR@colbangspec}%
        {\arabic{LWR@tabletotalLaTeXcols}}{\LWR@colparameter}%
    \LWR@traceinfo{bang \arabic{LWR@tabletotalLaTeXcols}: \LWR@colparameter!}%
}%
\let\LWR@colparameter\relax%
\global\booltrue{LWR@validtablecol}%
}
```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```
\newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```
\expandarg%
\StrChar{\LWR@origcolspec}%
    {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
\fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
\LWR@setexparray{LWR@colbeforespec}%
    {\arabic{LWR@tabletotalLaTeXcolsnext}}{\LWR@colparameter}%
\let\LWR@colparameter\relax%
\global\booltrue{LWR@validtablecol}%
}
```

\LWR@parseaftercolumn Handles <{text} columns.

```
\newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
\addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
\expandarg%
\StrChar{\LWR@origcolspec}%
    {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
\fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
\LWR@setexparray{LWR@colafterspec}%
    {\arabic{LWR@tabletotalLaTeXcols}}{\LWR@colparameter}%
\let\LWR@colparameter\relax%
\global\booltrue{LWR@validtablecol}%
}
```

\LWR@parsebarcolumn Handles vertical rules.

```
\newcommand*{\LWR@parsebarcolumn}{%
\LWR@traceinfo{LWR@parsebarcolumn}%
```

Remember the bar at this position:

```
\ifnumcomp{\value{LWR@tabletotalLaTeXcols}}{=}{0}%
{% left edge of the table:
    \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{leftedge}}%
    \ifdefstring{\LWR@tempone}{tvertbarl}%
    {\LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarldouble}}%
    {\LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarl}}%
}%
{% not at the left edge:
    \edef\LWR@tempone{%
        \LWR@getexparray{LWR@colbarspec}{\arabic{LWR@tabletotalLaTeXcols}}%
```

```
    }%
    \ifdefstring{\LWR@tempone}{tvertbarr}%
    {%
        \LWR@setexparray{LWR@colbarspec}%
            {\arabic{LWR@tabletotalLaTeXcols}}{tvertbarrdouble}%
    }%
    {%
        \LWR@setexparray{LWR@colbarspec}%
            {\arabic{LWR@tabletotalLaTeXcols}}{tvertbarr}%
    }%
}%
\global\booltrue{LWR@validtablecol}%
}
```

`\LWR@parsecoloncolumn` Handles vertical rules.

```
\newcommand*{\LWR@parsecoloncolumn}{%
\LWR@traceinfo{LWR@parsecoloncolumn}%
```

Remember the bar at this position:

```
\ifnumcomp{\value{LWR@tabletotalLaTeXcols}}{=}{0}%
{% left edge of the table:
    \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{leftedge}}%
    \ifdefstring{\LWR@tempone}{tvertbarldash}%
    {\LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarldoubledash}}%
    {\LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarldash}}%
}%
{% not at the left edge:
    \edef\LWR@tempone{%
        \LWR@getexparray{LWR@colbarspec}{\arabic{LWR@tabletotalLaTeXcols}}%
    }%
    \ifdefstring{\LWR@tempone}{tvertbarrdash}%
    {\LWR@setexparray{LWR@colbarspec}%
        {\arabic{LWR@tabletotalLaTeXcols}}{tvertbarrdoubledash}}%
    {\LWR@setexparray{LWR@colbarspec}%
        {\arabic{LWR@tabletotalLaTeXcols}}{tvertbarrdash}}%
}%
\global\booltrue{LWR@validtablecol}%
}
```

`\LWR@parsesemicoloncolumn` Handles vertical rules.

```
\newcommand*{\LWR@parsesemicoloncolumn}{%
```

Treat ; as a : column:

```
\LWR@parsecoloncolumn%
```

Skip the following width token:

```
\addtocounter{LWR@tablecolspecindex}{1}%
}
```

68.9 Parsing 'l', 'c', or 'r' columns

\LWR@parsenormalcolumn {⟨*thiscolumn*⟩}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
\newcommand*{\LWR@parsenormalcolumn}[1]{%
\addtocounter{LWR@tabletotalLaTeXcols}{1}%
\addtocounter{LWR@tabletotalLaTeXcolsnext}{1}%

\LWR@setexparray{LWR@tablecolspec}{\arabic{LWR@tabletotalLaTeXcols}}{#1}%

\LWR@traceinfo{normal column \arabic{LWR@tabletotalLaTeXcols}: #1}%
\LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalLaTeXcolsnext}}{}%
\LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalLaTeXcolsnext}}{}%
\LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tabletotalLaTeXcolsnext}}{}%
\LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tabletotalLaTeXcolsnext}}{}%
\LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tabletotalLaTeXcolsnext}}{}%
\global\booltrue{LWR@validtablecol}%
}
```

68.10 Parsing 'p', 'm', or 'b' columns

\LWR@parsepcolumn {⟨*thiscolumn*⟩} The width will be ignored.

```
\newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
\LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
\addtocounter{LWR@tablecolspecindex}{1}%
}
```

68.11 Parsing 'w' columns

\LWR@parsewcolumn The width will be ignored.

```
\newcommand*{\LWR@parsewcolumn}{%
\LWR@columnspeclookahead{1}%
\expandafter\LWR@parsenormalcolumn\expandafter{\LWR@strresulttwo}%
```

Skips the following width and alignment tokens:

```
\addtocounter{LWR@tablecolspecindex}{2}%
}
```

68.12 Parsing 'D' columns

From the **dcolumn** package.

\LWR@parseDcolumn {⟨*thiscolumn*⟩} The three parameters will be ignored.

```
\newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
\LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
\addtocounter{LWR@tablecolspecindex}{3}%
}
```

68.13 Parsing the column specifications

⚠ HTML css cannot exactly match the LaTeX concept of a baseline for a table row. Table 8 shows the LaTeX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 9 for details.

Table 9 describes how each kind of column is converted to HTML.

Bool \LWR@validtablecol True if found a valid table column type.

Table 8: Tabular baseline

l	p	m	b	r
l	par par par	mid mid mid	bot bot bot	r

Table 9: Tabular HTML column conversions

l, r, c: Converted to table cells without paragraph tags.
Uses CSS `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.

p: Converted to table cells with paragraph tags. Ref: Table 8, LaTeX places the top line of a parbox aligned with the rest of the text line, so CSS `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate LaTeX baselines.

m: With paragraph tags, CSS `vertical-align:middle`.

b: With paragraph tags, CSS `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.

P, M, B: Horizontally-centered versions.

S: Converted to 'r'. Ignores optional argument. From the **siunitx** package.

D: Converted to 'c'. From the **dcolumn** package.

@, !, >, <: One each, in that order.

|: Vertical rule.

Unknown: Converted to 'l'.

`\newcolumn`: Currently treated as unknown.

```
\newbool{LWR@validtablecol}
```

Bool `LWR@opttablecol` True if found a table column optional argument.

```
\newbool{LWR@opttablecol}
```

`\LWR@parsetablecols` {⟨*colspecs*⟩}

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one LaTeX column per entry. The final number of LaTeX columns in each row is stored in `LWR@tabletotalLaTeXcols`, which is the number of `&` and `\\` in each line, but which does not include `@`, `!`, `<`, `>` specifications in the count.

```
\newcommand*{\LWR@parsetablecols}[1]{%
\LWR@traceinfo{LWR@parsetablecols}%
```

Remember the original supplied column spec:

```
\renewcommand*{\LWR@origcolspec}{#1}%
```

Remove spaces:

```
\expandarg%
\StrSubstitute{\LWR@origcolspec}{ }{}[\LWR@origcolspec]%
```

The parsed column spec data array, `LWR@tablecolspec`, will be overwritten with new values.

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
\setcounter{LWR@tabletotalLaTeXcols}{0}%
\setcounter{LWR@tabletotalLaTeXcolsnext}{1}%
\LWR@setexparray{LWR@colatspec}{leftedge}{}%
\LWR@setexparray{LWR@colatspec}{1}{}%
\LWR@setexparray{LWR@colatspec}{2}{}%
\LWR@setexparray{LWR@colatspec}{3}{}%
\LWR@setexparray{LWR@colbangspec}{leftedge}{}%
\LWR@setexparray{LWR@colbangspec}{1}{}%
\LWR@setexparray{LWR@colbangspec}{2}{}%
\LWR@setexparray{LWR@colbangspec}{3}{}%
\LWR@setexparray{LWR@colbeforespec}{1}{}%
\LWR@setexparray{LWR@colbeforespec}{2}{}%
\LWR@setexparray{LWR@colbeforespec}{3}{}%
\LWR@setexparray{LWR@colafterspec}{1}{}%
```

```
\LWR@setexparray{LWR@colafterspec}{2}{}%
\LWR@setexparray{LWR@colafterspec}{3}{}%
\LWR@setexparray{LWR@colbarspec}{leftedge}{}%
\LWR@setexparray{LWR@colbarspec}{1}{}%
\LWR@setexparray{LWR@colbarspec}{2}{}%
\LWR@setexparray{LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
\setcounter{LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
\expandarg%
\StrLen{\LWR@origcolspec}[\LWR@strresult]%
\fullexpandarg%
\LWR@traceinfo{original column spec length: \LWR@strresult}%
\setcounter{LWR@tablecolspecwidth}{\LWR@strresult}%
```

Haven't seen any optional arguments so far

```
\global\boolfalse{LWR@opttablecol}%
```

Scan through the column specifications:

```
\whileboolexpr{%
    not test{%
        \ifnumcomp{\value{LWR@tablecolspecindex}}{>}%
            {\value{LWR@tablecolspecwidth}}%
    }%
}%
{%
```

Place the next single-character column type into `\LWR@strresult`:

```
\expandarg%
\StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@strresult]%
\LWR@traceinfo{position \arabic{LWR@tablecolspecindex}: \LWR@strresult}%
\fullexpandarg%
```

Not yet found a valid column type:

```
\global\boolfalse{LWR@validtablecol}%
```

Skip over any optional arguments, such as **siunitx** `S` column:

```
\IfStrEq{\LWR@strresult}{[}{\global\booltrue{LWR@opttablecol}}{}%
```

Throw away anything found inside the optional argument:

```
\ifbool{LWR@opttablecol}%
{}% inside an optional argument
{% not an optional tabular argument
```

Not inside an optional argument, so consider the column type:

```
\IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
\IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
\IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
\IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
\IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%

\IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{c}}{}%
\IfStrEq{\LWR@strresult}{s}{\LWR@parsenormalcolumn{c}}{}%

\IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
\IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
\IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
\IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
\IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
\IfStrEq{\LWR@strresult}{:}{\LWR@parsecoloncolumn}{}%
\IfStrEq{\LWR@strresult}{;}{\LWR@parsesemicoloncolumn}{}%

\IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
\IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
\IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

\IfStrEq{\LWR@strresult}{w}{\LWR@parsewcolumn}{}%
\IfStrEq{\LWR@strresult}{W}{\LWR@parsewcolumn}{}%
```

From the **dcolumn** package:

```
\IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the **tabularx** package. X column has no parameter, but will be given paragraph tags.

```
\IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions "`P`", "`M`", and "`B`":

```
\newcolumntype{P}[1]{>{\centering\arraybackslash}p{#1}}
```

```
\IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
\IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
\IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an `l` column:

```
\ifbool{LWR@validtablecol}{}{%
    \LWR@traceinfo{invalid column type: \LWR@strresult}%
    \LWR@parsenormalcolumn{l}%
}%
}% not an optional column argument
```

If read the closing bracket, no longer inside the optional argument:

```
\IfStrEq{\LWR@strresult}{]}{\global\boolfalse{LWR@opttablecol}}{}%
```

Move to the next character:

```
\addtocounter{LWR@tablecolspecindex}{1}%
}% whiledo
}%
```

68.14 colortbl and xparse tabular color support

These macros provide a minimal emulation of some **colortbl** macros which might appear between table cells. If **colortbl** is loaded, these macros will be replaced with functional versions.

For each of the HTML colors below, the text for the HTML color is set if requested, but the macro is empty if none has been set.

`\rownum` Reserve a counter register.

```
\@ifundefined{rownum}{\newcount\rownum}{}
```

`\@rowcolors` Emulated in case **xcolor** is not used.

```
\newcommand*{\@rowcolors}{}
```

`\@rowc@lors` Emulated in case **xcolor** is not used.

```
\newcommand*{\@rowc@lors}{}
```

\LWR@xcolorrowHTMLcolor Emulated **xcolor** row color.

```
6796 \newcommand*{\LWR@xcolorrowHTMLcolor}{}
```

\LWR@columnHTMLcolor HTMLstyle code for the column color.

```
6797 \def\LWR@columnHTMLcolor{}
```

\LWR@rowHTMLcolor HTMLstyle code for the row color.

```
6798 \def\LWR@rowHTMLcolor{}
```

\LWR@cellHTMLcolor HTMLstyle code for the cell color.

```
6799 \def\LWR@cellHTMLcolor{}
```

\LWR@ruleHTMLcolor HTMLstyle code for the cell color.

```
6800 \newcommand*{\LWR@ruleHTMLcolor}{}
```

\rowcolor [⟨*model*⟩] {⟨*color*⟩} [⟨*left overhang*⟩] [⟨*right overhang*⟩] Print version. The HTML version is in **lwarp-colortbl**. Used before starting a tabular data cell, thus \LWR@getmynexttoken.

```
6801 \newcommand*{\rowcolor}{\LWR@getmynexttoken}%
```

\arrayrulecolor [⟨*model*⟩] {⟨*color*⟩}

\arrayrulecolornexttoken [⟨*model*⟩] {⟨*color*⟩}

Print versions for use outside and inside a tabular:

```
6802 \newcommand{\arrayrulecolor}[2][named]{}
6803 \newcommand{\arrayrulecolornexttoken}[2][named]{\LWR@getmynexttoken}
```

\doublerulesepcolor [⟨*model*⟩] {⟨*color*⟩}

\doublerulesepcolornexttoken [⟨*model*⟩] {⟨*color*⟩}

Print versions for use inside and outside a tabular:

```
6804 \newcommand{\doublerulesepcolor}[2][named]{}
6805 \newcommand{\doublerulesepcolornexttoken}[2][named]{\LWR@getmynexttoken}
```

68.15 Starting a new row

\LWR@maybenewtablerow If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for `hline` or `tbrule` if necessary.

```
\newcommand*{\LWR@maybenewtablerow}
{%
\ifbool{LWR@startedrow}%
{}% started the row
{% not started the row
```

Remember that now have started the row:

```
    \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
    \global\booltrue{LWR@intabularmetadata}%
    \ifboolexpr{%
        test{\ifnumcomp{\value{LWR@hlines}}{>}{0}} or%
        test{\ifnumcomp{\value{LWR@hdashedlines}}{>}{0}}%
    }%
    {%
        \LWR@htmltag{tr class="hline" }%
        \LWR@orignewline%
    }%
    {% not doing hline
        \ifbool{LWR@doingtbrule}%
        {%
            \ifdefvoid{\LWR@ruleHTMLcolor}{%
                \LWR@htmltag{tr class="tbrule"}%
            }{%
                \LWR@htmltag{%
                    tr class="tbrule" % space
                    style="border-top: 1px solid % space
                        \LWR@origpound\LWR@ruleHTMLcolor "%
                }%
            }%
            \LWR@orignewline%
        }%
        {\LWR@htmltag{tr}\LWR@orignewline}%
    }% end of not doing hline
}% end of not started the row
}
```

68.16 Printing vertical bar tags

\LWR@printbartag {⟨*index*⟩}

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```
\newcommand*{\LWR@printbartag}[1]{%
\LWR@traceinfo{LWR@printbartag !#1!}%
\ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
{}% muting or empty
{% not muting
    \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{#1}}%
    \ifdefempty{\LWR@tempone}{}{ \LWR@tempone}%
}% not muting
\LWR@traceinfo{LWR@printbartag done}%
}
```

68.17 Printing at or bang tags

\LWR@printatbang {⟨*at -or- bang*⟩} {⟨*index*⟩}

```
\newcommand*{\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```
\xdef\LWR@atbangspec{\LWR@getexparray{LWR@col#1spec}{#2}}%
\LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```
\ifdefempty{\LWR@atbangspec}%
{}%
{% not empty
    \LWR@htmltag{%
        td class="td#1%
        \LWR@subaddcmidruletrim{}{}%
        \LWR@printbartag{#2}%
        "%
        \LWR@tdstartstyles%
        \LWR@addcmidrulewidth%
        \LWR@addcdashline%
        \LWR@addtabularrulecolors%
        \LWR@tdendstyles%
    }%
```

Create an empty cell if muting for the \bottomrule:

```
    \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
    {}%
    {\LWR@atbangspec}%
%
    \LWR@htmltag{/td}\LWR@orignewline%
    \global\booltrue{LWR@tabularcelladded}%
}% not empty
}%
```

\LWR@addleftmostbartag

```
\newcommand*{\LWR@addleftmostbartag}{%
\ifnumcomp{\value{LWR@tableLaTeXcolindex}}{=}{1}{%
    \LWR@printbartag{leftedge}%
}{}%
}
```

\LWR@tabularleftedge

```
\newcommand*{\LWR@tabularleftedge}{%
\ifnumcomp{\value{LWR@tableLaTeXcolindex}}{=}{1}%
{%
    \LWR@printatbang{at}{leftedge}%
    \LWR@printatbang{bang}{leftedge}%
}% left edge
{}% not left edge
}
```

68.18 Data opening tag

\LWR@thiscolspec Temporary storage.

```
\newcommand*{\LWR@thiscolspec}{}
```

\LWR@tabledatasinglecolumntag Print a table data opening tag with style for alignment and color.

```
\newcommand*{\LWR@tabledatasinglecolumntag}%
{%
\LWR@traceinfo{LWR@tabledatasinglecolumntag}%
\LWR@maybenewtablerow%
```

Don't start a new paragraph tag if have already started one:

```
\ifbool{LWR@intabularmetadata}%
{%
```

If have found the end of tabular command, do not create the next data cell:

```
    \ifbool{LWR@exitingtabular}{}%
    {% not exiting tabular
```

Print the `@` and `!` contents before first column:

```
        \LWR@tabularleftedge%
```

Fetch the current column's alignment character into `\LWR@strresult`:

```
        \xdef\LWR@strresult{%
            \LWR@getexparray{LWR@tablecolspec}{\arabic{LWR@tableLaTeXcolindex}}%
        }%
```

print the start of a new table data cell:

```
\LWR@traceinfo{LWR@tabledatasinglecolumntag: about to print td tag}%
        \LWR@htmltag{td class="td%
```

append this column's spec:

```
        \LWR@strresult%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add vertical bar tags.

```
        \LWR@addcmidruletrim%
        \LWR@addleftmostbartag%
        \LWR@printbartag{\arabic{LWR@tableLaTeXcolindex}}%
        "%
```

Add styles for rules, alignment:

```
        \LWR@tdstartstyles%
        \LWR@addcmidrulewidth%
        \LWR@addcdashline%


        \xdef\LWR@thiscolspec{%
            \LWR@getexparray{LWR@tablecolspec}{\arabic{LWR@tableLaTeXcolindex}}%
        }%
        \LWR@addformatwpalignment{\LWR@thiscolspec}%
```

Add styles for cell and rule colors:

```
        \LWR@addtabularrowcolor%
        \LWR@addtabularrulecolors%

        \LWR@tdendstyles%
        }% HTML td
        \LWR@traceinfo{LWR@tabledatasinglecolumntag: done printing td tag}%
```

If this is a `p`, `m`, `b`, or `X` column, allow paragraphs:

```
        \ifboolexpr{%
            test{ \ifdefstring{\LWR@strresult}{p} } or
            test{ \ifdefstring{\LWR@strresult}{m} } or
            test{ \ifdefstring{\LWR@strresult}{b} } or
            test{ \ifdefstring{\LWR@strresult}{P} } or
            test{ \ifdefstring{\LWR@strresult}{M} } or
            test{ \ifdefstring{\LWR@strresult}{B} } or
            test{ \ifdefstring{\LWR@strresult}{X} }
        }%
        {% allow pars
            \LWR@traceinfo{LWR@tabledatasinglecolumntag: about to LWR@startpars}%
            \global\booltrue{LWR@tableparcell}%
            \LWR@startpars%
            \LWR@traceinfo{LWR@tabledatasinglecolumntag: done with LWR@startpars}%
        }% allow pars
        {}% no pars
```

Print the > contents unless muted for the `\bottomrule`:

```
        \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
        {}%
        {%
            \LWR@getexparray{LWR@colbeforespec}{\arabic{LWR@tableLaTeXcolindex}}%
        }%
        \global\boolfalse{LWR@intabularmetadata}%
    }% not exiting tabular
}{}% in tabular metadata
\LWR@traceinfo{LWR@tabledatasinglecolumntag: done}%
}%
```

68.19 Midrules

`LWR@midrules` `LWR@midrules` is a data array (section 39) of columns each containing a non-zero width if a midrule should be created for this column.

LWR@trimlrules — LWR@trimlrules is a data array (section 39) of columns containing l if a midrule should be left trimmed for each column.

LWR@trimrrules — LWR@trimrrules is a data array (section 39) of columns containing r if a midrule should be right trimmed for each column.

LWR@cdashlines — LWR@cdashlines is a data array (section 39) of columns each containing a Y if an **arydshln** package "cdashed line" should be created for this column.

Ctr LWR@midrulecounter — Indexes across the LWR@midrules and LWR@trim<l/r>rules data arrays.

```
\newcounter{LWR@midrulecounter}
```

Len \LWR@heavyrulewidth — The default width of the rule.

```
\newlength{\LWR@heavyrulewidth}
\setlength{\LWR@heavyrulewidth}{.08em}
```

Len \LWR@lightrulewidth — The default width of the rule.

```
\newlength{\LWR@lightrulewidth}
\setlength{\LWR@lightrulewidth}{.05em}
```

Len \LWR@cmidrulewidth — The default width of the rule.

```
\newlength{\LWR@cmidrulewidth}
\setlength{\LWR@cmidrulewidth}{.03em}
```

Len \LWR@thiscmidrulewidth — The width of the next rule, defaulting to \LWR@cmidrulewidth.

If not \LWR@cmidrulewidth, a style will be used to generate the custom width.

Assigned from the LWR@midrules array.

```
\newlength{\LWR@thiscmidrulewidth}
\setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}
```

\LWR@clearmidrules — Start new midrules. Called at beginning of tabular and also at \\.

Clears all LWR@midrules and LWR@trimrules markers for this line.

```
\newcommand*{\LWR@clearmidrules}
{%
\setcounter{LWR@midrulecounter}{1}%
\whileboolexpr{%
    not test{%
        \ifnumcomp{\value{LWR@midrulecounter}}{>}%
            {\value{LWR@tabletotalLaTeXcols}}%
```

```
    }%
}%
{%
    \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{0pt}%
    \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
    \LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
    \LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
    \LWR@setexparray{LWR@cdashlines}{\arabic{LWR@midrulecounter}}{N}%
    \addtocounter{LWR@midrulecounter}{1}%
}%
}
```

`\LWR@subcmidrule` {⟨*width*⟩} {⟨*trim*⟩} {⟨*leftcolumn*⟩} {⟨*rightcolumn*⟩}

Marks `LWR@midrules` data array elements to be non-zero widths from left to right columns. Also marks trimming for the L and/or R columns.

`LWR@doingcmidrule` is set to force an empty row at the end of the tabular to create the rule.

```
\newcommand*{\LWR@subcmidrule}[4]{%
\setcounter{LWR@midrulecounter}{#3}%
\whileboolexpr{%
    not test {%
        \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}%
    }%
}%
{%
    \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
    \addtocounter{LWR@midrulecounter}{1}%
}% whiledo
\IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
\IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
\global\booltrue{LWR@doingcmidrule}%
}
```

`\LWR@docmidrule` [⟨*width*⟩] (⟨*trim*⟩) {⟨*leftcolumn-rightcolumn*⟩}

Marks `LWR@midrules` array elements to be a non-zero width from left to right columns. Also marks trimming for the L and/or R columns.

```
\NewDocumentCommand{\LWR@docmidrule}
    {O{\LWR@cmidrulewidth} D(){} >{\SplitArgument{1}{-}}m}
    {\LWR@subcmidrule{#1}{#2}#3}
```

`\LWR@subcdashline` {⟨*leftcolumn*⟩} {⟨*rightcolumn*⟩}

Marks `LWR@cdashlines` data array elements to be `Y` from left to right columns.

LWR@doingcmidrule is set to force an empty row at the end of the tabular to create the rule.

```
\newcommand*{\LWR@subcdashline}[2]{%
\setcounter{LWR@midrulecounter}{#1}%
\whileboolexpr{%
    not test {%
        \ifnumcomp{\value{LWR@midrulecounter}}{>}{#2}%
    }%
}%
{%
    \LWR@setexparray{LWR@cdashlines}{\arabic{LWR@midrulecounter}}{Y}%
    \addtocounter{LWR@midrulecounter}{1}%
}% whiledo
\global\booltrue{LWR@doingcmidrule}%
}
```

\LWR@docdashline {⟨*leftcolumn-rightcolumn*⟩}

Marks LWR@cdashlines data array elements to be Y from left to right columns.

```
\NewDocumentCommand{\LWR@docdashline}
    {>{\SplitArgument{1}{-}}m}%
    {%
        \LWR@subcdashline#1%
    }
```

Used to compute margins, tabular trims, column offsets:

```
\newlength{\LWR@templengthone}
\newlength{\LWR@templengththwo}
\newlength{\LWR@templengthththree}
\newcounter{LWR@tempcountone}
```

Used to add a style to a table data cell:

```
\newbool{LWR@tdhavecellstyle}
```

\LWR@tdstartstyles Begins possibly adding a table data cell style.

```
\newcommand*{\LWR@tdstartstyles}{\global\boolfalse{LWR@tdhavecellstyle}}
```

\LWR@tdaddstyle Starts adding a table data cell style.

```
\newcommand*{\LWR@tdaddstyle}{%
\ifbool{LWR@tdhavecellstyle}%
```

```
{; }%
{ style="}%
\booltrue{LWR@tdhavecellstyle}%
}
```

`\LWR@tdendstyles` Finishes possibly adding a table data cell style. Prints the closing quote.

```
\newcommand*{\LWR@tdendstyles}{%
\ifbool{LWR@tdhavecellstyle}{%
    "%
    \global\boolfalse{LWR@tdhavecellstyle}%
}{}%
}
```

`\LWR@subaddcmidruletrim` {⟨*lefttrim*⟩} {⟨*righttrim*⟩} Adds a `\cmidrule` with optional trim.

```
\newcommand*{\LWR@subaddcmidruletrim}[2]{%
\setlength{\LWR@templengthone}{%
        \LWR@getexparray{LWR@midrules}{\arabic{LWR@tableLaTeXcolindex}}%
}%
\ifdimcomp{\LWR@templengthone}{>}{0pt}%
{%
```

Print the class with left and right trim letters appended:

```
    \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
    \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
}%
{%
    \setlength{\LWR@thiscmidrulewidth}{0pt}%
}%
}
```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```
\newcommand*{\LWR@addcmidruletrim}{%
\LWR@subaddcmidruletrim%
{\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tableLaTeXcolindex}}}%
{\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@tableLaTeXcolindex}}}%
}
```

`\LWR@addrulewidth` {⟨*thiswidth*⟩} {⟨*defaultwidth*⟩}

If not default width, add a custom style with width and color depending on `thiswidth`.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
\newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`, or if a color is being used:

```
\ifboolexpr{%
    test{\ifdimcomp{#1}{=}{0pt}} or
    (
        ( test{\ifdimcomp{#1}{=}{#2}} and not bool{FormatWP} )
        and ( test {\ifdefvoid{\LWR@ruleHTMLcolor}} )
    )
}%
{}% default width and color
{% custom width and/or color
```

Ensure that the width is wide enough to display in the browser:

```
    \LWR@forceminwidth{#1}%
```

Begin adding another style:

```
    \LWR@tdaddstyle%
```

The style itself:

```
    border-top:\LWR@printlength{\LWR@atleastonept} solid %
```

If default gray, the darkness of the color depends on the thickness of the rule:

```
    \ifdefvoid{\LWR@ruleHTMLcolor}{%
        \ifdimcomp{#1}{<}{\LWR@lightrulewidth}%
        {\LWR@origpound{}A0A0A0}%
        {% lightrule or heaver
            \ifdimcomp{#1}{<}{\LWR@heavyrulewidth}%
            {\LWR@origpound{}808080}%
            {black}%
        }% lightrule or heavier
    }{%
        \LWR@origpound\LWR@ruleHTMLcolor%
    }
}% custom width and/or color
}
```

\LWR@addcmidrulewidth Adds a `style` for the rule width.

Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
\newcommand{\LWR@addcmidrulewidth}{%
\LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
}
```

\LWR@addcdashline Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
\newcommand{\LWR@addcdashline}{%
\edef\LWR@tempone{%
    \LWR@getexparray{LWR@cdashlines}{\arabic{LWR@tableLaTeXcolindex}}%
}%
\ifdefstring{\LWR@tempone}{Y}{%
    \LWR@tdaddstyle%
    border-top: 1pt dashed %
    \ifdefvoid{\LWR@ruleHTMLcolor}%
        {black}%
        {\LWR@origpound\LWR@ruleHTMLcolor}%
}{}%
}
```

\LWR@WPcell {⟨*text-align*⟩} {⟨*vertical-align*⟩}

```
\newcommand*{\LWR@WPcell}[2]{%
\LWR@tdaddstyle%
\LWR@print@mbox{text-align:#1}; \LWR@print@mbox{vertical-align:#2}%
}
```

\LWR@addformatwpalignment If `FormatWP`, adds a `style` for the alignment.

Must be placed between \LWR@tdstartstyles and \LWR@tdendstyles.

```
\newcommand*{\LWR@addformatwpalignment}[1]{%
\ifbool{FormatWP}{%
\IfSubStr{#1}{l}{\LWR@WPcell{left}{middle}}{}%
\IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
\IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%
\IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
\IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
\IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
\IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
\IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
\IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
}{}%
}
```

68.20 Cell colors

\LWR@addtabularrowcolor Adds a cell's row color style, if needed.

No color is added for the final row of empty cells which finishes each `tabular`.

```
\newcommand*{\LWR@addtabularrowcolor}{%
\ifbool{LWR@tabularmutemods}{}{%
    \ifdefvoid{\LWR@rowHTMLcolor}{%
        \ifdefvoid{\LWR@xcolorrowHTMLcolor}{}%
        {% xcolor row color
            \LWR@tdaddstyle%
            background:\LWR@origpound\LWR@xcolorrowHTMLcolor%
        }%
    }%
    {% explicit row color
        \LWR@tdaddstyle%
        background:\LWR@origpound\LWR@rowHTMLcolor%
    }%
}%
}
```

\LWR@addtabularhrulecolor Adds a cell's horizontal rule color style, if needed.

```
\newcommand*{\LWR@addtabularhrulecolor}{%
```

If either form of horizontal rule is requested:

```
\ifboolexpr{%
    test{\ifnumcomp{\value{LWR@hlines}}{>}{0}} or%
    test{\ifnumcomp{\value{LWR@hdashedlines}}{>}{0}} or%
    bool{LWR@doingtbrule}%
}{%
```

If there is a no custom color:

```
    \ifdefvoid{\LWR@ruleHTMLcolor}%
    {%
        \ifnumcomp{\value{LWR@hlines}}{>}{1}%
        {%
            \LWR@tdaddstyle%
            border-top: 4px double%
        }{% else
        \ifnumcomp{\value{LWR@hdashedlines}}{>}{1}%
        {%
            \LWR@tdaddstyle%
            border-top: 2px dashed%
        }{% else
```

```
        \ifnumcomp{\value{LWR@hdashedlines}}{=}{1}%
        {%
            \LWR@tdaddstyle%
            border-top: 1px dashed%
        }{}}}%
```

If no color and not doubled or dashed, then add nothing, since a simpler rule is the default.

```
    }%
```

If there is a custom color:

```
    {%
        \ifnumcomp{\value{LWR@hlines}}{>}{1}%
        {%
            \LWR@tdaddstyle%
            border-top: 4px double \LWR@origpound\LWR@ruleHTMLcolor%
        }{% else
        \ifnumcomp{\value{LWR@hdashedlines}}{>}{1}%
        {%
            \LWR@tdaddstyle%
            border-top: 2px dashed \LWR@origpound\LWR@ruleHTMLcolor%
        }{% else
        \ifnumcomp{\value{LWR@hdashedlines}}{=}{1}%
        {%
            \LWR@tdaddstyle%
            border-top: 1px dashed \LWR@origpound\LWR@ruleHTMLcolor%
        }{% else
            \LWR@tdaddstyle%
            border-top: 1px solid \LWR@origpound\LWR@ruleHTMLcolor%
        }}}%
    }%
}{}%
}
```

`\LWR@addtabularrulecolors` Adds a cell's rule color styles, if needed.

No color is added for the final row of empty cells which finishes each `tabular`.

```
\newcommand*{\LWR@addtabularrulecolors}{%
```

Custom horizonal rule color:

```
\LWR@addtabularhrulecolor%
```

No vertical rules if finishing the `tabular` with a row of empty cells:

```
\ifbool{LWR@tabularmutemods}{}{%
```

If at the leftmost cell, possibly add a leftmost vertical rule:

```
    \ifnumequal{\value{LWR@tableLaTeXcolindex}}{1}{%
```

Fetch the left edge's vertical bar specification:

```
        \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{leftedge}}%
```

Add a custom style if a vertical bar was requested:

```
        \ifdefstring{\LWR@tempone}{tvertbarl}{%
                \LWR@tdaddstyle%
                border-left: 1px solid % space
                    \LWR@origpound\LWR@vertruleHTMLcolor%
        }{}%
        \ifdefstring{\LWR@tempone}{tvertbarldouble}{%
                \LWR@tdaddstyle%
                border-left: 4px double % space
                    \LWR@origpound\LWR@vertruleHTMLcolor%
        }{}%
        \ifdefstring{\LWR@tempone}{tvertbarldash}{%
                \LWR@tdaddstyle%
                border-left: 1px dashed % space
                    \LWR@origpound\LWR@vertruleHTMLcolor%
        }{}%
        \ifdefstring{\LWR@tempone}{tvertbarldoubledash}{%
                \LWR@tdaddstyle%
                border-left: 2px dashed % space
                    \LWR@origpound\LWR@vertruleHTMLcolor%
        }{}%
    }{}%
```

Possibly add a right vertical rule for this cell:

```
    \edef\LWR@tempone{%
        \LWR@getexparray{LWR@colbarspec}{\arabic{LWR@tableLaTeXcolindex}}%
    }%
    \ifdefstring{\LWR@tempone}{tvertbarr}{%
```

Add a custom style if a vertical bar was requested:

```
            \LWR@tdaddstyle%
            border-right: 1px solid \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifdefstring{\LWR@tempone}{tvertbarrdouble}{%
            \LWR@tdaddstyle%
```

```
            border-right: 4px double \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifdefstring{\LWR@tempone}{tvertbarrdash}{%
            \LWR@tdaddstyle%
            border-right: 1px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifdefstring{\LWR@tempone}{tvertbarrdoubledash}{%
            \LWR@tdaddstyle%
            border-right: 2px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
}%
}
```

Ctr `LWR@cellcolordepth` Counts how many cell color `<div>`s were added to the current tabular data cell.

```
\newcounter{LWR@cellcolordepth}
```

`\LWR@subaddtabularcellcolor` {⟨*HTML color*⟩}

```
\newcommand*{\LWR@subaddtabularcellcolor}[1]{%
\LWR@htmltag{div class="cellcolor" style="%
    background:\LWR@origpound{}{}#1 %
" }%
\addtocounter{LWR@cellcolordepth}{1}%
}
```

`\LWR@addtabularcellcolor` Adds a cell color style, if needed.

```
\newcommand*{\LWR@addtabularcellcolor}{%
\ifdefvoid{\LWR@cellHTMLcolor}%
{%
    \ifdefvoid{\LWR@rowHTMLcolor}%
    {%
        \ifdefvoid{\LWR@xcolorrowHTMLcolor}%
        {%
            \ifdefvoid{\LWR@columnHTMLcolor}%
            {}%
            {\LWR@subaddtabularcellcolor{\LWR@columnHTMLcolor}}%
        }%
        {\LWR@subaddtabularcellcolor{\LWR@xcolorrowHTMLcolor}}%
    }%
    {\LWR@subaddtabularcellcolor{\LWR@rowHTMLcolor}}%
}%
{\LWR@subaddtabularcellcolor{\LWR@cellHTMLcolor}}%
}
```

68.21 Multicolumns

68.21.1 Parsing multicolumns

```
\newcounter{LWR@tablemulticolswidth}
```

Indexes into the multicolumn specification:

```
\newcounter{LWR@tablemulticolspos}
```

Remembers multicolumn vertical rules if found in the column spec.

```
\newcounter{LWR@mcolvertbarsl}
\newcounter{LWR@mcolvertbarsr}
\newcounter{LWR@mcolvertbarsldash}
\newcounter{LWR@mcolvertbarsrdash}
\newbool{LWR@mcolvertbaronleft}%
```

`\LWR@printmccoltype` {⟨*colspec*⟩} Print any valid column type found. Does not print `@`, `!`, `>`, or `<` columns or their associated tokens.

This is printed as part of the table data tag's `class`.

```
\newcommand*{\LWR@printmccoltype}[1]{%
\LWR@traceinfo{lwr@printmccoltype -#1-}%
```

Get one token of the column spec:

```
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
\IfStrEq{\LWR@strresult}{l}{l}{}%
\IfStrEq{\LWR@strresult}{c}{c}{}%
\IfStrEq{\LWR@strresult}{r}{r}{}%
\IfStrEq{\LWR@strresult}{p}{p}{}%
\IfStrEq{\LWR@strresult}{m}{m}{}%
\IfStrEq{\LWR@strresult}{b}{b}{}%
\IfStrEq{\LWR@strresult}{P}{P}{}%
\IfStrEq{\LWR@strresult}{M}{M}{}%
\IfStrEq{\LWR@strresult}{B}{B}{}%

\IfStrEq{\LWR@strresult}{w}{w}{}%
\IfStrEq{\LWR@strresult}{W}{W}{}%

\IfStrEq{\LWR@strresult}{S}{c}{}%
\IfStrEq{\LWR@strresult}{s}{c}{}%
```

```
\IfStrEq{\LWR@strresult}{X}{p}{}%

\IfStrEq{\LWR@strresult}{|}%
{%
    \ifbool{LWR@mcolvertbaronleft}%
        {\addtocounter{LWR@mcolvertbarsl}{1}}% left edge
        {\addtocounter{LWR@mcolvertbarsr}{1}}% not left edge
}%
{%
    \IfStrEq{\LWR@strresult}{:}%
    {%
        \ifbool{LWR@mcolvertbaronleft}%
            {\addtocounter{LWR@mcolvertbarsldash}{1}}% left edge
            {\addtocounter{LWR@mcolvertbarsrdash}{1}}% not left edge
    }%
    {%
        \IfStrEq{\LWR@strresult}{;}%
        {%
            \ifbool{LWR@mcolvertbaronleft}%
                {\addtocounter{LWR@mcolvertbarsldash}{1}}% left edge
                {\addtocounter{LWR@mcolvertbarsrdash}{1}}% not left edge
        }%
        {\boolfalse{LWR@mcolvertbaronleft}}%
    }%
}%
\LWR@traceinfo{lwr@printmccoltype done}%
}
```

`\LWR@multicolpartext` {⟨*num parameters*⟩} Print the data with paragraph tags, advance to bypass the given number of parameters.

```
\newcommand*{\LWR@multicolpartext}[1]{%
\LWR@startpars%
\LWR@multicoltext%
\addtocounter{LWR@tablemulticolspos}{#1}%
\LWR@stoppars%
}
```

`\LWR@multicolother` {⟨*colspec*⟩} For `@`, `!`, `>`, `<`, print the next token without paragraph tags:

```
\newcommand*{\LWR@multicolother}[1]{%
\addtocounter{LWR@tablemulticolspos}{1}%
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
\LWR@strresult%
```

A valid column data type was found:

```
\global\booltrue{LWR@validtablecol}%
}
```

`\LWR@multicolskip` Nothing to print for this column type.

```
\newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
\global\booltrue{LWR@validtablecol}%
}
```

`\LWR@printmccoldata` {⟨*colspec*⟩} Print the data for any valid column type found.

```
\newcommand*{\LWR@printmccoldata}[1]{%
\LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
\global\boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
\StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles `@`, `>`, `<` as it comes to them.

```
\IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{D}{%
\addtocounter{LWR@tablemulticolspos}{3}% skip parameters
\LWR@multicoltext%
}{}%

\IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext{0}}{}%
\IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext{0}}{}%
\IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext{0}}{}%
\IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext{0}}{}%
\IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext{0}}{}%
\IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext{0}}{}%

\IfStrEq{\LWR@strresult}{w}{\LWR@multicolpartext{3}}{}%
\IfStrEq{\LWR@strresult}{W}{\LWR@multicolpartext{3}}{}%
```

```
\IfStrEq{\LWR@strresult}{S}{\LWR@multicoltext}{}%
\IfStrEq{\LWR@strresult}{s}{\LWR@multicoltext}{}%

\IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext{0}}{}%
\IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
\IfStrEq{\LWR@strresult}{:}{\LWR@multicolskip}{}%
\IfStrEq{\LWR@strresult}{;}{%
    \LWR@multicolskip%
    \addtocounter{LWR@tablemulticolspos}{1}% skip parameter
}{}%
\IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
\IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%
```

If an invalid column type:

```
\ifbool{LWR@validtablecol}{}{\LWR@multicoltext}%
```

Tracing:

```
\LWR@traceinfo{lwr@printmccoldata done}%
}
```

`\parsemulticolumnalignment` {⟨*1: colspec*⟩} {⟨*2: printresults*⟩}

Scan the multicolumn specification and execute the printfunction for each entry.

Note that the spec for a `p{spec}` column, or `@`, `>`, `<`, is a token list which will NOT match `l`, `c`, `r`, or `p`.

```
\newcommand*{\LWR@parsemulticolumnalignment}[2]{%
\setcounter{LWR@tablemulticolspos}{1}%
\StrLen{#1}[\LWR@strresult]%
\setcounter{LWR@tablemulticolswidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
\whileboolexpr{%
    not test {%
        \ifnumcomp{\value{LWR@tablemulticolspos}}{>}%
            {\value{LWR@tablemulticolswidth}}%
    }%
}%
{%
```

Execute the assigned print function for each token in the column spec:

```
#2{#1}%
```

Move to the next token in the column spec:

```
\addtocounter{LWR@tablemulticolspos}{1}%
}%
}
```

68.21.2 Multicolumn factored code

`\LWR@addmulticolvertrulecolor`

```
\newcommand*{\LWR@addmulticolvertrulecolor}{%
```

No vertical rules if finishing the `tabular` with a row of empty cells:

```
\ifbool{LWR@tabularmutemods}{}{%
```

Left side:

```
    \ifnumcomp{\value{LWR@mcolvertbarsl}}{=}{1}{%
        \LWR@tdaddstyle%
        border-left: 1px solid \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifnumcomp{\value{LWR@mcolvertbarsl}}{>}{1}{%
        \LWR@tdaddstyle%
        border-left: 4px double \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifnumcomp{\value{LWR@mcolvertbarsldash}}{=}{1}{%
        \LWR@tdaddstyle%
        border-left: 1px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifnumcomp{\value{LWR@mcolvertbarsldash}}{>}{1}{%
        \LWR@tdaddstyle%
        border-left: 2px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
```

Right side:

```
    \ifnumcomp{\value{LWR@mcolvertbarsr}}{=}{1}{%
        \LWR@tdaddstyle%
        border-right: 1px solid \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifnumcomp{\value{LWR@mcolvertbarsr}}{>}{1}{%
        \LWR@tdaddstyle%
        border-right: 4px double \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
```

```
    \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{=}{1}{%
        \LWR@tdaddstyle%
        border-right: 1px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
    \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{>}{1}{%
        \LWR@tdaddstyle%
        border-right: 2px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
    }{}%
}%
}
```

```
\newcommand{\LWR@multicoltext}{}
```

To find multicolumn right trim:

```
\newcounter{LWR@lastmulticolumn}
```

`\LWR@domulticolumn` `[`⟨*1: vpos*⟩`]` `[`⟨*2: #rows*⟩`]` {⟨*3: numLaTeXcols*⟩} {⟨*4: numHTMLcols*⟩} {⟨*5: colspec*⟩} {⟨*6: text*⟩}

```
\NewDocumentCommand{\LWR@domulticolumn}{o o m m m +m}{%
\LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
\renewcommand{\LWR@multicoltext}{%
#6%
\global\booltrue{LWR@validtablecol}%
}%
```

Compute the rightmost column to be included. This is used to create the right trim.

```
\setcounter{LWR@lastmulticolumn}{\value{LWR@tableLaTeXcolindex}}%
\addtocounter{LWR@lastmulticolumn}{#3}%
\addtocounter{LWR@lastmulticolumn}{-1}%
```

Row processing:

```
\LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
\LWR@htmltag{td colspan="#4" %
```

```
\IfValueT{#2}{ % rows?
rowspan="#2" %
```

```
\IfValueT{#1}{% vpos?
\ifstrequal{#1}{b}{style="\LWR@print@mbox{vertical-align:bottom}" }{}%
\ifstrequal{#1}{t}{style="\LWR@print@mbox{vertical-align:top}" }{}%
}% vpos?
}% rows?

 class="td%
```

Print the column type and vertical bars:

```
\setcounter{LWR@mcolvertbarsl}{0}%
\setcounter{LWR@mcolvertbarsr}{0}%
\setcounter{LWR@mcolvertbarsldash}{0}%
\setcounter{LWR@mcolvertbarsrdash}{0}%
\booltrue{LWR@mcolvertbaronleft}%
\LWR@parsemulticolumnalignment{#5}{\LWR@printmccoltype}%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag.

If this position had a "Y" then add "rule" for a horizontal rule:

```
\LWR@subaddcmidruletrim%
{\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tableLaTeXcolindex}}}%
{\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%
```

Also add vertical bar class.

```
\ifnumcomp{\value{LWR@mcolvertbarsl}}{=}{1}{ tvertbarl}{}%
\ifnumcomp{\value{LWR@mcolvertbarsl}}{>}{1}{ tvertbarldouble}{}%
\ifnumcomp{\value{LWR@mcolvertbarsr}}{=}{1}{ tvertbarr}{}%
\ifnumcomp{\value{LWR@mcolvertbarsr}}{>}{1}{ tvertbarrdouble}{}%
\ifnumcomp{\value{LWR@mcolvertbarsldash}}{=}{1}{ tvertbarldash}{}%
\ifnumcomp{\value{LWR@mcolvertbarsldash}}{>}{1}{ tvertbarldoubledash}{}%
\ifnumcomp{\value{LWR@mcolvertbarsrdash}}{=}{1}{ tvertbarrdash}{}%
\ifnumcomp{\value{LWR@mcolvertbarsrdash}}{>}{1}{ tvertbarrdoubledash}{}%
```

Close the class tag's opening quote: " NOT A TYPO

```
%

\LWR@tdstartstyles%

\LWR@addtabularrowcolor%

\LWR@addcmidrulewidth%
\LWR@addcdashline%
\LWR@addtabularhrulecolor%
```

```
\LWR@addmulticolvertrulecolor%
\LWR@addformatwpalignment{#5}%
\LWR@tdendstyles%
}% end of the opening table data tag
\global\boolfalse{LWR@intabularmetadata}%
\LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
}
```

68.21.3 Multicolumn

`\LWR@htmlmulticolumn` {⟨*numcols*⟩} {⟨*alignment*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
{%
```

Figure out how many extra HTML columns to add for `@` and `!` columns:

```
\LWR@tabularhtmlcolumns{\arabic{LWR@tableLaTeXcolindex}}{#1}
```

Create the multicolumn tag:

```
\LWR@domulticolumn{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#3}%
```

Move to the next LaTeX column:

```
\addtocounter{LWR@tableLaTeXcolindex}{#1}%
\addtocounter{LWR@tableLaTeXcolindex}{-1}%
```

Skip any trailing `@` or `!` columns for this cell:

```
\global\booltrue{LWR@skipatbang}%
}
```

68.21.4 Longtable captions

longtable captions use `\multicolumn`.

Bool `LWR@starredlongtable` Per the **caption** pacakge, step the counter if `longtable*`.

```
\newbool{LWR@starredlongtable}
\boolfalse{LWR@starredlongtable}
```

Per the **caption** package. User-redefinable float type.

```
\providecommand*{\LTcaptype}{table}
```

\LWR@longtabledatacaptiontag * [⟨*toc entry*⟩] {⟨*caption*⟩}

```
\NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
{%
```

Remember the latest name for `\nameref`:

```
\IfValueTF{#2}{% optional given?
    \ifblank{#2}% optional empty?
    {\LWR@setlatestname{#3}}% empty
    {\LWR@setlatestname{#2}}% given and non-empty
}% optional given
{\LWR@setlatestname{#3}}% no optional
```

Create a multicolumn across all the columns:

Figure out how many extra HTML columns to add for `@` and `!` columns found between the first and the last column:

```
\LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalLaTeXcols}}
```

Create the multicolumn tag:

```
\LWR@domulticolumn{\arabic{LWR@tabletotalLaTeXcols}}%
    {\arabic{LWR@tabhtmlcoltotal}}%
    {P}%
{% \LWR@domulticolumn
\IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
{% yes star
    \LWR@figcaption%
    #3%
    \endLWR@figcaption%
}%
{% No star:
```

Not the star version:

Don't step the counter if `\caption[]{A caption.}`

```
    \ifbool{LWR@starredlongtable}%
    {%
        \ifblank{#2}% TOC entry
        {}%
        {%
            \refstepcounter{\LTcaptype}%
```

```
            \protected@edef\@currentlabel{%
                \@nameuse{p@\LTcaptype}\@nameuse{the\LTcaptype}%
            }%
        }%
    }{}%
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
    \LWR@figcaption%
    \@nameuse{fnum@\LTcaptype}\CaptionSeparator#3%
    \endLWR@figcaption%
```

See if an optional caption was given:

```
    \ifblank{#2}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
    {}%
```

If the optional caption was given, but might only be `[]`:

```
    {% TOC entry not empty
        \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is `[]`:

```
        {% No TOC entry
            \addcontentsline%
            {\@nameuse{ext@\LTcaptype}}%
            {\LTcaptype}%
            {%
            \protect\numberline%
            {\@nameuse{p@\LTcaptype}\@nameuse{the\LTcaptype}}%
            {\ignorespaces #3\protect\relax}%
            }%
        }% end of No TOC entry
```

The optional caption has text enclosed:

```
        {% yes TOC entry
            \addcontentsline%
            {\@nameuse{ext@\LTcaptype}}%
            {\LTcaptype}%
            {%
            \protect\numberline%
            {\@nameuse{p@\LTcaptype}\@nameuse{the\LTcaptype}}%
            {\ignorespaces #2\protect\relax}%
```

```
            }%
        }% end of yes TOC entry
    }% end of TOC entry not empty
}% end of no star
```

Skip any trailing @ or ! columns for this cell:

```
\global\booltrue{LWR@skipatbang}%
}% end of \LWR@domulticolumn

\addtocounter{LWR@tableLaTeXcolindex}{\arabic{LWR@tabletotalLaTeXcols}}
\addtocounter{LWR@tableLaTeXcolindex}{-1}

}
```

68.21.5 Counting HTML tabular columns

The LaTeX specification for a table includes a number of columns separated by the & character. These columns differ in content from line to line. Additional virtual columns may be specified by the special @ and ! columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output, @ and ! columns are placed into their own tabular columns. Thus, a LaTeX \multicolumn command may span several additional @ and ! columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```
\newcounter{LWR@tabhtmlcolindex}
\newcounter{LWR@tabhtmlcolend}
\newcounter{LWR@tabhtmlcoltotal}
```

\LWR@subtabularhtmlcolumns {⟨*index*⟩}

Factored from \LWr@tabularhtmlcolumns, which follows.

```
\newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the @ specification for this column:

```
    \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the @ specification is not empty, add to the count:

```
    \ifdefempty{\LWR@atbangspec}%
        {}%
        {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
    \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{#1}}%
    \ifdefempty{\LWR@atbangspec}%
        {}%
        {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
}
```

`\LWR@tabularhtmlcolumns` {⟨*starting LaTeX column*⟩} {⟨*number LaTeX columns*⟩}

Compute the total number of HTML columns being spanned, considering the starting LaTeX table column and the number of LaTeX tabular columns being spanned. Any @ and ! columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter LWR@tabhtmlcoltotal.

```
\newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the LaTeX span, to which additional @ and ! columns may be added:

```
\setcounter{LWR@tabhtmlcolindex}{#1}%
\setcounter{LWR@tabhtmlcoltotal}{#2}%
\setcounter{LWR@tabhtmlcolend}{#1}%
\addtocounter{LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
\ifnumcomp{\value{LWR@tabhtmlcolindex}}{=}{1}{%
    \LWR@subtabularhtmlcolumns{leftedge}%
}{}%
```

Walk across the LaTeX columns looking for @ and ! columns:

```
\whileboolexpr{%
    test {%
        \ifnumcomp{\value{LWR@tabhtmlcolindex}}{<}{\value{LWR@tabhtmlcolend}}%
    }%
}%
{%
    \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
    \addtocounter{LWR@tabhtmlcolindex}{1}%
}% whiledo
}
```

```
\end{warpHTML}
```

68.22 Multirow if not loaded

A default defintion in case **multirow** is not loaded. This is used during table parsing.

```
\begin{warpHTML}
\newcommand{\multirow}[2][c]{}
\end{warpHTML}
```

68.23 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 285 for the HTML versions.

for HTML & PRINT:

```
\begin{warpall}
```

`\multicolumnrow` {⟨*1:cols*⟩} {⟨*2:halign*⟩} [⟨*3:vpos*⟩] {⟨*4:numrows*⟩} [⟨*5:bigstruts*⟩] {⟨*6:width*⟩} [⟨*7:fixup*⟩] {⟨*8:text*⟩}

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:
https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex

`\AtBeginDocument` to adjust after the user may have loaded **multirow**, which requires several tests to determine which version is loaded and thus which options are available.

```
\AtBeginDocument{
```

`\@ifundefined{@xmultirow}` determines if **multirow** was never loaded.

Null action if not loaded:

```
\@ifundefined{@xmultirow}
{
\DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
    {}%
}% no version of multirow was loaded
{% \@xmultirow defined, so some version of multirow was loaded
```

`\@ifpackageloaded{multirow}` determines if v2.0 or later of **multirow** was used, which included the `\ProvidesPackage` macro.

The print version:

```
\@ifpackageloaded{multirow}{% v2.0 or newer
\@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
{% v2.0+:
\DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
    {\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}[#5]{#6}[#7]{#8}}}%
}
{% loaded but older, probably not executed:
\DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
    {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
}
}% packageloaded{multirow}
```

If not `\@ifpackageloaded{multirow}` but `\@xmultirow` is defined, then this must be v1.6 or earlier, which did not `\ProvidesPackage{multirow}`, and did not have the `vposn` option.

```
{% v1.6 or older did not \ProvidePackage
\DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
    {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
    {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
}

}% \@ifundefined{@xmultirow}

\providecommand*{\multicolumnrow}{\LWR@print@multicolumnrow}

}% AtBeginDocument

\end{warpall}
```

68.24 Utility macros inside a table

for HTML output:

```
\begin{warpHTML}
```

Used to prevent opening a tabular data cell if the following token is one which does not create tabular data:

```
\newcommand*{\LWR@donothing}{}
```

In case **array** is not loaded:

```
\let\firsthline\relax
```

```
\let\lasthline\relax
\newcommand*{\firsthline}{}
\newcommand*{\lasthline}{}
```

In case **bigdelim** is not loaded:

```
\newcommand*{\ldelim}{}
\newcommand*{\rdelim}{}
```

```
\end{warpHTML}
```

68.25 Special-case tabular markers

for HTML & PRINT:

```
\begin{warpall}
```

`\TabularMacro` Place this just before inserting a custom macro in a table data cell. Doing so tells **lwarp** not to automatcally start a new HTML table data cell yet. See section 9.9.

```
\newcommand*{\TabularMacro}{}
```

```
\end{warpall}
```

`\ResumeTabular` Used to resume tabular entries after resuming an environment.

⚠ `tabular` inside another environment

When creating a new environment which contains a `tabular` environment, **lwarp**'s emulation of the tabular does not automatically resume when the containing enviroment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\StopDefiningTabulars
```

for HTML output:

```
\begin{warpHTML}
```

```
\newcommand*{\ResumeTabular}{%
\global\boolfalse{LWR@exitingtabular}%
\global\boolfalse{LWR@tabularmutemods}%
\LWR@getmynexttoken%
}
```

```
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
```

```
\newcommand*{\ResumeTabular}{}
```

```
\end{warpprint}
```

68.26 Checking for a new table cell

for HTML output:

```
\begin{warpHTML}
```

Bool `LWR@exitingtabular` When `\end` is found, turns off the next opening data tag.

```
\newbool{LWR@exitingtabular}
```

Bool `LWR@tabularmutemods` Mutes HTML output for `@`, `!`, < and >.

This is used while printing the final row to generate `\bottomrules`.

```
\newbool{LWR@tabularmutemods}
```

`\LWR@tabledatacolumntag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
\newcommand*{\LWR@tabledatacolumntag}%
{%
\LWR@traceinfo{LWR@tabledatacolumntag}%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
\global\let\LWR@mynextaction\LWR@tabledatasinglecolumntag%
```

If exiting the tabular:

```
\ifdefequal{\LWR@mynexttoken}{\end}%
    {\global\booltrue{LWR@exitingtabular}}{}%
```

`longtable` can have a caption in a cell

```
\ifdefequal{\LWR@mynexttoken}{\caption}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
\ifdefequal{\LWR@mynexttoken}{\multicolumn}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
\ifdefequal{\LWR@mynexttoken}{\multirow}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
\ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
\ifdefequal{\LWR@mynexttoken}{\noalign}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

If an `\mrowcell`, this is a cell to be skipped over:

```
\ifdefequal{\LWR@mynexttoken}{\mrowcell}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

If an `\mcolrowcell`, this is a cell to be skipped over:

```
\ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\hline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\firsthline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\lasthline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\toprule}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\midrule}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

```
\ifdefequal{\LWR@mynexttoken}{\cmidrule}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\morecmidrules}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\specialrule}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\cline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\bottomrule}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\rowcolor}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\arrayrulecolor}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\doublerulesepcolor}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\ldelim}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\rdelim}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

For **arydshln**:

```
\ifdefequal{\LWR@mynexttoken}{\hdashline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\cdashline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%

\ifdefequal{\LWR@mynexttoken}{\firsthdashline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

```
\ifdefequal{\LWR@mynexttoken}{\lasthdashline}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

Ignore an empty line between rows:

```
\ifdefequal{\LWR@mynexttoken}{\par}%
    {\global\let\LWR@mynextaction\LWR@donothing}{}%
```

No action for an \end token.

Add similar to the above for any other non-data tokens which might appear in the table.

Start the new table cell if was not any of the above:

```
\LWR@traceinfo{LWR@tabledatacolumntag: about to do mynext}%
\LWR@mynextaction%
\LWR@traceinfo{LWR@tabledatacolumntag: done}%
}
```

```
\end{warpHTML}
```

68.27 \mrowcell

for HTML & PRINT:

```
\begin{warpall}
```

\mrowcell ⚠

The user must insert \mrowcell into any \multirow cells which must be skipped. This command has no action during print output.

```
\newcommand*{\mrowcell}{}
```

```
\end{warpall}
```

68.28 \mcolrowcell

for HTML & PRINT:

```
\begin{warpall}
```

\mcolrowcell ⚠

The user must insert \mcolrowcell into any \multicolumnrow cells which must be skipped. This command has no action during print output.

```
\newcommand*{\mcolrowcell}{}
```

```
\end{warpall}
```

68.29 HTML `tabular` environment

for HTML output:
```
\begin{warpHTML}
```

These are default defininitions in case **booktabs** is not loaded, and are not expected to used, but must exist as placeholders. They are pre-deleted in case **memoir** has already loaded **booktabs**.

```
\LetLtxMacro\toprule\relax
\LetLtxMacro\midrule\relax
\LetLtxMacro\cmidrule\cline
\LetLtxMacro\bottomrule\relax
\LetLtxMacro\addlinespace\relax
\LetLtxMacro\morecmidrules\relax
\LetLtxMacro\specialrule\relax

\newcommand*{\toprule}[1][]{\hline}
\newcommand*{\midrule}[1][]{\hline}
\LetLtxMacro\cmidrule\cline
\newcommand*{\bottomrule}[1][]{\hline}
\newcommand*{\addlinespace}[1][]{}
\newcommand*{\morecmidrules}{}
\newcommand*{\specialrule}[3]{\hline}
```

`\noalign` {⟨*text*⟩} Redefined for use inside `tabular`.

```
\LetLtxMacro\LWR@orignoalign\noalign

\newcommand{\LWR@tabularnoalign}[1]{%
\begingroup%
\global\advance\rownum\m@ne%
\renewcommand*{\LWR@xcolorrowHTMLcolor}{}%
\multicolumn{\value{LWR@tabletotalLaTeXcols}}{l}{#1} \\
\endgroup%
% \@rowc@lors%
\LWR@getmynexttoken%
}
```

`\LWR@HTMLhline` The definition of `\hline` depends on whether **tabls** has been loaded. If so, optional space below the line may be specified, but will be ignored.

```
\AtBeginDocument{
\@ifpackageloaded{lwarp-tabls}
{
\newcommand*{\LWR@HTMLhline}[1][]{%
    \ifbool{FormatWP}%
    {\LWR@docmidrule{1-\arabic{LWR@tabletotalLaTeXcols}}}%
```

```
    {\addtocounter{LWR@hlines}{1}}%
    \LWR@getmynexttoken}%
}
{
\newcommand*{\LWR@HTMLhline}{%
    \ifbool{FormatWP}%
    {\LWR@docmidrule{1-\arabic{LWR@tabletotalLaTeXcols}}}%
    {\addtocounter{LWR@hlines}{1}}%
    \LWR@getmynexttoken}%
}
}% AtBeginDocument
```

`\LWR@HTMLcline` {⟨*columns*⟩}

```
\NewDocumentCommand{\LWR@HTMLcline}{m}%
{\LWR@docmidrule{#1}\LWR@getmynexttoken}%
```

`\LWR@nullifyNoAutoSpacing` For **babel-french**, turn off auto spacing at the start of the tabular, then nullify the autospacing commands inside the tabular, since they were not compatible with the tabular column parsing code, which uses **xstring**.

```
\AtBeginDocument{
\@ifundefined{frenchbsetup}%
{% no babel-french
    \newcommand*{\LWR@nullifyNoAutoSpacing}{}
}% no babel-french
{% yes babel-french
    \newcommand*{\LWR@nullifyNoAutoSpacing}{%
        \NoAutoSpacing%
        \renewcommand*{\NoAutoSpacing}{}%
        \renewcommand*{\LWR@FBcancel}{}%
    }
}% yes babel-french
}% AtBeginDocument
```

Env `LWR@tabular` [⟨*vertposition*⟩] {⟨*colspecs*⟩}

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since **siunitx** might redefine tabular in the user's document.

```
\StartDefiningTabulars

\newenvironment*{LWR@tabular}[2][]
{%
\LWR@traceinfo{LWR@tabular started}%
\addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
\global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```
\setcounter{LWR@hlines}{0}%
\setcounter{LWR@hdashedlines}{0}%
\global\boolfalse{LWR@doingtbrule}%
\global\boolfalse{LWR@doingcmidrule}%
```

For **babel-french**, turn off auto spacing one time, then nullify the autospacing commands since were not compatible with the tabular parsing code.

```
\LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the `@` and `!` columns.

```
\global\boolfalse{LWR@exitingtabular}%
\global\boolfalse{LWR@tabularmutemods}%
```

Create the `table` tag:

```
\global\booltrue{LWR@intabularmetadata}%
\LWR@traceinfo{LWR@tabular: About to LWR@forecenewpage.}%
\LWR@forcenewpage
\LWR@htmlblocktag{table}%
```

Parse the table columns:

```
\LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
\LWR@stoppars%
```

Track column #:

```
\setcounter{LWR@tableLaTeXcolindex}{1}%
```

Have not yet added data in this column:

```
\global\boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
\LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
\LetLtxMacro{\\}{\LWR@tabularendofline}%
```

The following adjust for **colortbl**.

```
\LetLtxMacro\arrayrulecolor\arrayrulecolornexttoken%
\LetLtxMacro\doublerulesepcolor\doublerulesepcolornexttoken%
\gdef\LWR@columnHTMLcolor{}%
\gdef\LWR@rowHTMLcolor{}%
\gdef\LWR@cellHTMLcolor{}%
\@rowcolors%
```

The vertical rules are set to the color active at the start of the tabular. `\arrayrulecolor` will then affect horizontal rules inside the tabular, but not the vertical rules.

```
\edef\LWR@vertruleHTMLcolor{\LWR@ruleHTMLcolor}%
```

Tracking the depth of cell color `<div>`s:

```
\setcounter{LWR@cellcolordepth}{0}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with `\LWR@getmynextoken` to see if the next token might create a new data cell:

The optional parameter for `\hline` supports the **tabls** package.

```
\LWR@traceinfo{LWR@tabular: redefining macros}%
\LetLtxMacro\noalign\LWR@tabularnoalign%
\LetLtxMacro\hline\LWR@HTMLhline%
\LetLtxMacro\cline\LWR@HTMLcline%

\DeclareDocumentCommand{\hdashline}{o}{%
    \ifbool{FormatWP}%
        {\LWR@docdashline{1-\arabic{LWR@tabletotalLaTeXcols}}}%
        {\addtocounter{LWR@hdashedlines}{1}}%
    \LWR@getmynexttoken%
}%

\DeclareDocumentCommand{\cdashline}{m}{%
    \LWR@docdashline{##1}\LWR@getmynexttoken%
}%

\DeclareDocumentCommand{\firsthdashline}{o}{%
```

```
    \ifbool{FormatWP}%
        {\LWR@docdashline{1-\arabic{LWR@tabletotalLaTeXcols}}}%
        {\addtocounter{LWR@hdashedlines}{1}}%
    \LWR@getmynexttoken%
}%

\DeclareDocumentCommand{\lasthdashline}{o}{%
    \ifbool{FormatWP}%
        {\LWR@docdashline{1-\arabic{LWR@tabletotalLaTeXcols}}}%
        {\addtocounter{LWR@hdashedlines}{1}}%
    \LWR@getmynexttoken%
}%
```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use `\LWR@getmynexttoken`.

```
\renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
\renewcommand*{\mrowcell}{%
    \LWR@maybenewtablerow%
    \LWR@tabularleftedge%
    \global\booltrue{LWR@skippingmrowcell}%
}%
\renewcommand*{\mcolrowcell}{%
    \LWR@maybenewtablerow%
    \global\booltrue{LWR@skippingmcolrowcell}%
}%
\LetLtxMacro\caption\LWR@longtabledatacaptiontag%
```

Reset for new processing:

```
\global\boolfalse{LWR@tableparcell}%
\global\boolfalse{LWR@skippingmrowcell}%
\global\boolfalse{LWR@skippingmcolrowcell}%
\global\boolfalse{LWR@skipatbang}%
\global\boolfalse{LWR@emptyatbang}%
```

Set `&` for its special meaning inside the tabular:

```
\StartDefiningTabulars%
\protected\gdef&{\LWR@tabularampersand}%
```

Nest one level deeper of tabular paragraph handling:

```
\addtocounter{LWR@tabularpardepth}{1}%
```

Look ahead for a possible table data cell:

```
\LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
```

```
\LWR@getmynexttoken%
}%
```

Ending the environment:

```
{%
\LWR@traceinfo{LWR@tabular ending}%
```

Unnest one level of tabular paragraph handling:

```
\addtocounter{LWR@tabularpardepth}{-1}%
\ifboolexpr{%
    test {%
        \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{<}{\value{LWR@tabletotalLaTeXcols}}
    } or %
    (%
        bool{LWR@intabularmetadata} and%
        not bool{LWR@tabularcelladded} and%
        test {%
            \ifnumcomp{\value{LWR@tableLaTeXcolindex}}{=}{\value{LWR@tabletotalLaTeXcols}}%
        }%
    )%
}%
{%
    \LWR@tabularfinishrow%
}%
{%
    \LWR@closetabledatacell%
}%
\LWR@htmlblocktag{/tr}%
```

xcolor row color support:

```
\@rowc@lors%
```

```
\LWR@htmlblocktag{/table}%
\global\boolfalse{LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
\addtocounter{LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
\protected\gdef&{\LWR@origampmacro}%
\StopDefiningTabulars%
\LWR@traceinfo{LWR@tabular finished ending}%
}
```

```
7852
7853 \StopDefiningTabulars

7854 \end{warpHTML}
```

69 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The **zref** package is used to remember section name, file, and lateximage depth and number for each label.

Table 10 shows the data structures related to cross-referencing.

for HTML output:
```
7855 \begin{warpHTML}
```

69.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
7856 \providecommand*{\@currentlabelname}{}
```

`\LWR@stripperiod` {⟨*text*⟩} [⟨.⟩]

Removes a trailing period.

```
7857 \def\LWR@stripperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` {⟨*object name*⟩}

Removes `\label`, strips any final period, and remembers the result.

```
7858 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove `\label` and other commands from the name, the strip any final period. See **zref-titleref** and **gettitlestring**.

```
7859 \GetTitleStringExpand{#1}%
7860 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
```

Table 10: Cross-referencing data structures

Original LaTeX: (print and HTML)

`\refstepcounter`**:** Steps the couunter and sets `\@currentlabel`.

`\@currentlabel`**:** `\p@<ctr>\the<ctr>` Updated by `\refstepcounter`.

`\label`**:** Writes to the `.aux` file:
`\newlabel{<label>}{{\@currentlabel}{\thepage}}`

`\newlabel`**:** When the `.aux` file is read, sets `\r@<label>`.

`\r@<label>`**:** Set to: `{{\@currentlabel}{\thepage}}`

`\ref`**:** Returns the first part of `\r@<label>`.

`\pageref`**:** Returns the second part of `\r@<label>`.

Added by lwarp: (HTML only)

`\label`**:** Adds HTML tags (section 69.3), plus `\splabel` data (section 69.2):

zLWR@name: The section name for this label.
zLWR@htmlfilenumer: The filenumber or name for this label.
zLWR@lateximagedepth: The `lateximagedepth` for this label.
zLWR@lateximagenumber: The `lateximagenumber` for this label.

`\nameref`**:** Emualted from **hyperref** for **lwarp**. See section 69.4.

`\ref` **and** `\nameref`**:** Adds HTML tags. See section 69.4.

Added by amsmath: (print and HTML)

`\label`**:** Execution is delayed until the math environment is completed.

`\ltx@label`**:** LaTeX `\label`, (HTML: patched by **lwarp**,) later patched by **cleveref**.

Added by cleveref: (print and HTML)

`\refstepcounter:` Added: sets `\cref@currentlabel`.

`\cref@currentlabel`**:** (`<type>=<ctr>` unless an alias is used):
`[<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>}` Also see section 55.4 for use with footnotes.

`\label`**:** Writes to the `.aux` file:
`\newlabel{<label>@cref}{{\cref@currentlabel}{\thepage}}`

`\newlabel`**:** (Unchanged.) When the `.aux` file is read, sets `\r@<label>@cref`.

`\r@<label>@cref`**:** Set to: `{{\cref@currentlabel}{\thepage}}`

Utility functions: See `\cref@getlabel`, `\cref@gettype`, `\cref@getcounter`, `\cref@getprefix`.

Cross-referencing names: `\crefname` and `\Crefname` assign human-readable names for references to this counter type.

Additionally patched by lwarp: (HTML only)

`\cref`**, etc.:** Modified for **lwarp**. See section 83.

`\label` **inside math:** See section 75.7.1.

Footnotes: See `\noteentry` in section 55.4.

```
\edef\@currentlabelname{%
\expandafter\LWR@stripperiod\@currentlabelname%
\ltx@empty.\ltx@empty\@nil%
}%
}
```

69.2 Zref setup

See:
http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference

Create a new property list called special:

```
\zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
\zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
\zref@newprop{zLWR@htmlfilenumber}{%
\ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}%
}%
```

Additional properties for lateximages:

```
\zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
\zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}
```

`zLWR@htmlfilenumber` property holds the file number or name

Add a `LWR@htmlfilenumber` property, and lateximage properties to special:

```
\zref@addprop{special}{zLWR@name}
\zref@addprop{special}{zLWR@htmlfilenumber}
\zref@addprop{special}{zLWR@lateximagedepth}
\zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
\newcommand*{\LWR@spref}[2]{%
\zref@extractdefault{#1}{#2}{??}%
}
```

\LWR@nameref {⟨*label*⟩} Returns the section name for this label:

```
\newcommand*{\LWR@nameref}[1]{%
\LWR@spref{#1}{zLWR@name}%
}
```

\LWR@htmlfileref {⟨*label*⟩} Returns the file number or name for this label:

```
\newcommand*{\LWR@htmlfileref}[1]{%
```

DO NOT USE \LWR@traceinfo HERE! Will be expanded.

```
\LWR@spref{#1}{zLWR@htmlfilenumber}%
}
```

\LWR@lateximagedepthref {⟨*label*⟩} Returns the lateximagedepth for this label:

```
\newcommand*{\LWR@lateximagedepthref}[1]{%
\LWR@spref{#1}{zLWR@lateximagedepth}%
}
```

\LWR@lateximagenumberref {⟨*label*⟩} Returns the lateximagenumber for this label:

```
\newcommand*{\LWR@lateximagenumberref}[1]{%
\LWR@spref{#1}{zLWR@lateximagenumber}%
}
```

\LWR@splabel {⟨*label*⟩} Sanitize the name and then creates the label:

```
\newcommand*{\LWR@splabel}[1]{%
\LWR@traceinfo{LWR@splabel !#1!}%
\LWR@setlatestname{\@currentlabelname}%
\zref@labelbylist{#1}{special}%
}
```

69.3 Labels

\LWR@sublabel {⟨*label*⟩} Creates an HTML id tag.

\detokenize is used to allow underscores in the labels.

```
\newcommand*{\LWR@sublabel}[1]{%
\LWR@traceinfo{LWR@sublabel !#1!}%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{}%
{% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored…)

```
    \LWR@sanitize{#1}%
    \ifbool{LWR@doingstartpars}%
    {% pars allowed
        \ifbool{LWR@doingapar}%
        {% par started
            \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
        }% par started
        {% par not started
            \LWR@stoppars%
            \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
            \LWR@startpars%
        }% par not started
    }% pars allowed
    {% pars not allowed
        \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
    }% pars not allowed
}% not lateximage
}
```

`\LWR@new@label` `(`⟨*bookmark*⟩`)` `{`⟨*label*⟩`}` `[`⟨*type*⟩`]`

`\label` during HTML output when not in SVG math mode, removing extra spaces around the label, as done by regular LaTeX `\label`.

The is also used during a `lateximage`, including SVG math, since the special label handling is required, but `\LWR@sublabel` does not generate HTML tags inside a `lateximage`.

clevereref later encases this to add its own cross-referencing.

The optional ⟨*bookmark*⟩ is per the **memoir** class, and is ignored.

The optional ⟨*type*⟩ is per the **ntheorem** package, and is ignored.

```
\NewDocumentCommand{\LWR@new@label}{d() m o}{%
\LWR@traceinfo{LWR@new@label: starting}%
\LWR@traceinfo{LWR@new@label: !#2!}%
% \@bsphack%
```

Create a traditional LaTeX label, as modified by **cleveref**:

```
\LWR@orig@label{#2}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```
\LWR@traceinfo{LWR@new@label: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
\LWR@traceinfo{LWR@new@label: LWR@thisfilename is !\LWR@thisfilename!}%
\LWR@traceinfo{LWR@new@label: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
\LWR@splabel{#2}%
\LWR@sublabel{#2}%
% \@esphack%
\LWR@traceinfo{LWR@new@label: done}%
}
```

69.4 References

`\LWR@startref` {⟨*label*⟩} (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```
\newcommand*{\LWR@startref}[1]
{%
\edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
\LWR@sanitize{#1}%
\LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
\LWR@htmltag{a href="%
\LWR@traceinfo{LWR@startref B}%
\LWR@print@mbox{\LWR@htmlrefsectionfilename{#1}}%
\LWR@traceinfo{LWR@startref C}%
\LWR@origpound%
```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```
\LWR@traceinfo{LWR@startref D: !#1!}%
\ifthenelse{\equal{\LWR@lidref}{??}}%
```

"??" if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```
{%
    \LWR@traceinfo{LWR@startref D0: ??}%
    ??%
}%
```

If `LWR@lateximagedepth` is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```
{%
    \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
    \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
    {%
        \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
        lateximage\LWR@lateximagenumberref{#1}%
    }%
    {%
        \LWR@traceinfo{LWR@startref D3}%
```

`\detokenize` is used to allow underscores in the labels:

```
        \LWR@print@mbox{\LWR@sanitized}%
    }%
}%
\LWR@traceinfo{LWR@startref E}%
```

Closing quote:

```
"}%
\LWR@traceinfo{LWR@startref F}%
}
```

`\LWR@subnewref` `{`⟨*label*⟩`}` `{`⟨*label or sub@label*⟩`}`

Factored for the **subfig** package. Uses the original label for the hyper-reference, but prints its own text, such as "`1(b)`".

```
\NewDocumentCommand{\LWR@subnewref}{m m}{%
\LWR@traceinfo{LWR@subnewref #1 #2}%
\LWR@startref{#1}%
\LWR@print@ref{#2}%
\LWR@htmltag{/a}%
}
```

`\ref` * `{`⟨*label*⟩`}` `\ref` is redefined to `\LWR@HTML@ref`, except inside the text part of a `\hyperref`, where it is redefined to `\LWR@ref@ignorestar`.

\LWR@HTML@ref * {⟨*label*⟩} Create an internal document reference link, or without a link if starred per **hyperref**.

```
\NewDocumentCommand{\LWR@HTML@ref}{s m}{%
\LWR@traceinfo{LWR@HTML@ref !#2!}%
\IfBooleanTF{#1}%
{\LWR@print@ref{#2}}%
{\LWR@subnewref{#2}{#2}}%
}

\LWR@formatted{ref}
```

\LWR@ref@ignorestar * {⟨*label*⟩} For use inside \hyperref. Ignores the star, then uses the original \ref.

```
\NewDocumentCommand{\LWR@ref@ignorestar}{s m}{%
    \LWR@print@ref{#2}%
}
```

\pagerefPageFor Text for page references.

```
\newcommand*{\pagerefPageFor}{see }
```

\pageref * {⟨*label*⟩} Create an internal document reference, or just the unlinked number if starred, per **hyperref**.

```
\NewDocumentCommand{\LWR@new@pageref}{s m}{%
\IfBooleanTF{#1}%
{(\pagerefPageFor\LWR@print@ref{#2})}%
{(\cpageref{#2})}%
}
```

\nameref {⟨*label*⟩}

```
\newrobustcmd*{\nameref}[1]{%
\LWR@traceinfo{nameref}%
\LWR@startref{#1}%
\LWR@traceinfo{nameref B}%
\LWR@nameref{#1}%
\LWR@traceinfo{nameref C}%
\LWR@htmltag{/a}%
\LWR@traceinfo{nameref: done}%
}
```

\Nameref {⟨*label*⟩} In print, adds the page number. In HTML, does not.

```
\LetLtxMacro\Nameref\nameref
```

69.5 Hyper-references

⚠ Note that the code currently only sanitizes the underscore character. Additional characters should be rendered inert as well. See the `hyperref.sty` definition of `\gdef\hyper@normalise` for an example.

Pkg `hyperref`

⚠ Do not tell other packages that **hyperref** is emulated. Some packages patch various commands if **hyperref** is present, which will probably break something, and the emulation already handles whatever may be emulated anyhow.

⚠ Any reference to `\usepackage{hyperref}` must be placed inside a `warpprint` environment.

```
% DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
% \EmulatesPackage{hyperref}[2015/08/01]% Disabled.  Do not do this.
```

Emulates **hyperref**:

`\@currentHref` Added to support **backref**.

```
\AtBeginDocument{
\def\@currentHref{%
autopage-\theLWR@currentautosec%
}
}
```

`\LWR@subhyperref` {⟨*URL*⟩}

Starts a link for `\LWR@hrefb`. A group must have been opened first, with nullified catcodes. The text name is printed afterwards, after the group is closed and catcodes restored.

```
\NewDocumentCommand{\LWR@subhyperref}{m}{%
\LWR@traceinfo{LWR@subhyperref !#1!}%
\LWR@sanitize{#1}%
\LWR@htmltag{%
    a href="\LWR@sanitized" %
    target="\_{}blank"\LWR@orignewline%
}%
}
```

`\LWR@subhyperreftext` {⟨*text*⟩}

Finishes the hyperref for \LWR@hrefb. Catcodes must have been restored already. To be used after \LWR@subhyperref, and after its group has been closed.

```
\newcommand{\LWR@subhyperreftext}[1]{%
#1%
\LWR@htmltag{/a}%
\LWR@ensuredoingapar%
}
```

\LWR@subhyperrefclass {⟨*URL*⟩} {⟨*text*⟩} {⟨*htmlclass*⟩}

```
\NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
\LWR@htmltag{%
    a href="%
        \begingroup\@sanitize#1\endgroup%
    " %
    class="#3"\LWR@orignewline%
}%
#2%
\LWR@htmltag{/a}%
\LWR@ensuredoingapar%
}
```

\href [⟨*options*⟩] {⟨*URL*⟩}

Create a link with accompanying text:

```
\DeclareDocumentCommand{\LWR@hrefb}{O{} m}{%
\LWR@ensuredoingapar%
\LWR@subhyperref{#2}%
\endgroup% restore catcodes
\LWR@subhyperreftext%
}

\newrobustcmd*{\href}{%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@hrefb%
}
```

\nolinkurl {⟨*URL*⟩}

Print the name of the link without creating the link:

```
\newcommand*{\LWR@nolinkurlb}[1]{%
\LWR@ensuredoingapar%
\def\LWR@templink{#1}%
\@onelevel@sanitize\LWR@templink%
\LWR@templink%
\endgroup%
}

\newrobustcmd*{\nolinkurl}{%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@nolinkurlb%
}
```

`\url` {⟨*URL*⟩}

Create a link whose text name is the address of the link.

The **url** package may redefine `\url`, so it is `\let` to `\LWR@urlahere` and also redefined by **lwarp-url**.

```
\DeclareDocumentCommand{\LWR@urlb}{m}{%
\LWR@ensuredoingapar%
\def\LWR@templink{#1}%
\@onelevel@sanitize\LWR@templink%
\href{\LWR@templink}{\LWR@templink}%
\endgroup%
}

\newrobustcmd*{\url}{%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@urlb%
}
```

`\LWR@subinlineimage` [⟨*alttag*⟩] {⟨*class*⟩} {⟨*filename*⟩} {⟨*extension*⟩} {⟨*style*⟩}

```
\newcommand*{\LWR@subinlineimage}[5][]{%
\ifblank{#1}%
{\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}%
```

```
{\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"}}%
}

\end{warpHTML}
```

Table 11: Float data structures

For each <type> of float (figure, table, etc.) there exists the following:

counter <type>: A counter called <type>, such as `figure`, `table`.

\<type>name: Name. `\figurename` prints "Figure", etc.

\ext@<type>: File extension. `\ext@figure` prints "lof", etc.

\fps@<type>: Placement.

\the<type>: Number. `\thetable` prints the number of the table, etc.

\p@<type>: Parent's number. Prints the number of the [within] figure, etc.

\fnum@<type>: Prints the figure number for the caption.
`\<type>name \the<type>`, "Figure 123".

\<type>: Starts the float environment. `\figure` or `\begin{figure}`

\end<type>: Ends the float environment. `\endfigure` or `\end{figure}`

\tf@<ext>: The LaTeX file identifier for the output file.

LWR@have<type>: A boolean remembering whether a `\listof` was requested for a float of this type.

File with extension lo<f,t,a-z>: An output file containing the commands to build the `\listof<type>` "table-of-contents" structure.

Cross-referencing names: For **cleveref**'s `\cref` and related, `\crefname` and `\Crefname` assign human-readable names for references to this float type.

70 Floats

Floats are supported, although partially through emulation.

Table 11 shows the data structure associated with each <type> of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the **babel** package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: "Figure 123 – Caption text".

70.1 Float environment

for HTML output:

```
\begin{warpHTML}
```

`\LWR@floatbegin` {⟨*type*⟩} [⟨*placement*⟩] Begins a `\newfloat` environment.

```
\NewDocumentCommand{\LWR@floatbegin}{m o}{%
\ifbool{FormatWP}{\newline}{}%
\LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
\addtocounter{LWR@thisautoid}{1}%
\booltrue{LWR@freezethisautoid}%

\begingroup%
```

Settings while inside the environment:

```
\LWR@print@raggedright%
```

Open an HTML `figure` tag. The figure is assigned a `class` equal to its type, and another class according to the **float** package style, if used. Note that `\csuse` returns an empty string if `\LWR@floatstyle@<type>` is note defined.

```
\LWR@htmltag{%
    figure id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}" % space
    class="#1 \@nameuse{LWR@floatstyle@#1}"%
}%
\ifbool{FormatWP}{%
    \LWR@orignewline%
    \LWR@BlockClassWP{}{}{wp#1}%
}{}%
```

Update the caption type:

```
\renewcommand*{\@captype}{#1}%
\caption@settype{#1}%
```

Mark the float for a word processor conversion:

```
\LWR@startpars%
\ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%

=== begin #1 ===

}{}%
```

Look for \centering, etc:

```
\LWR@futurenonspacelet\LWR@mynexttoken\LWR@floatalignment%
}
```

For **koma-script**. The following does not work for tables.

```
\AtBeginDocument{
\@ifpackageloaded{tocbasic}{
\appto\figure@atbegin{%
\LWR@futurenonspacelet\LWR@mynexttoken\LWR@floatalignment%
}
}{}
}
```

\@float
\@dlbfloat

Support packages which create floats directly.

```
\let\@float\LWR@floatbegin
\let\@dblfloat\LWR@floatbegin
```

\LWR@floatend

Ends a \newfloat environment.

```
\newcommand*{\LWR@floatend}{%
```

If saw a \centering, finish the center environment:

```
\LWR@endfloatalignment%
```

Mark the float end for a word processor conversion:

```
\ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%

=== end ===

}{}%
\LWR@stoppars%
```

Close an HTML figure tag:

```
\ifbool{FormatWP}{\endLWR@BlockClassWP}{}%
\LWR@htmlelementend{figure}%
\endgroup%
\boolfalse{LWR@freezethisautoid}%
\LWR@startpars%
\ifbool{FormatWP}{\newline}{}%
}
```

\end@float
\end@dlbfloat
Support packages which create floats directly.

```
\let\end@float\LWR@floatend
\let\end@dblfloat\LWR@floatend
```

70.2 Float tracking

Ctr LWR@thisautoid
A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

```
\newcounter{LWR@thisautoid}
```

Ctr LWR@thisautoidWP
A sequential counter for all word processor conversion s. This is used to convince LibreOffice to form a frame around this element.

```
\newcounter{LWR@thisautoidWP}
```

Bool LWR@freezethisautoid
Prevents multiple increments of \LWR@thisautoid inside a float.

```
\newbool{LWR@freezethisautoid}
\boolfalse{LWR@freezethisautoid}
```

\LWR@newautoidanchor
Adds a new <autoid> anchor.

```
\newcommand*{\LWR@newautoidanchor}{%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{}%
{
    \ifbool{LWR@freezethisautoid}{}{%
        \addtocounter{LWR@thisautoid}{1}%
        \LWR@htmltag{a id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}"}%
            \LWR@htmltag{/a}%
    }%
}
}
```

\@captype
Remembers which float type is in use.

```
\newcommand*{\@captype}{}
```

\LWR@floatalignmentname
Set to center, flushleft, or flushright if saw \centering, \raggedright, or \raggedleft.

```
\newcommand*{\LWR@floatalignmentname}{}
```

\LWR@floatalignment If sees a \centering, \raggedleft, or \raggedright, creates a center, flushright, or flushleft environment.

```
\newcommand*{\LWR@floatalignment}{%
\ifdefstrequal{\LWR@mynexttoken}{\centering}{%
    \center%
    \renewcommand*{\LWR@floatalignmentname}{center}%
}{}%
\ifdefstrequal{\LWR@mynexttoken}{\raggedright}{%
    \flushleft%
    \renewcommand*{\LWR@floatalignmentname}{flushleft}%
}{}%
\ifdefstrequal{\LWR@mynexttoken}{\raggedleft}{%
    \flushright%
    \renewcommand*{\LWR@floatalignmentname}{flushright}%
}{}%
}
```

\LWR@endfloatalignment Closes an environment from \LWR@floatalignment.

```
\newcommand*{\LWR@endfloatalignment}{%
\ifdefvoid{\LWR@floatalignmentname}{}{\@nameuse{end\LWR@floatalignmentname}}%
\renewcommand*{\LWR@floatalignmentname}{}%
}
```

70.3 Caption inside a float environment

\CaptionSeparator How to separate the float number and the caption text.

```
\AtBeginDocument{\providecommand*{\CaptionSeparator}{:~}}
```

\@makecaption {⟨*name and num*⟩} {⟨*text*⟩}

Prints the float type and number, the caption separator, and the caption text.

```
\AtBeginDocument{\renewcommand{\@makecaption}[2]{%
    \LWR@traceinfo{@makecaption}%
    #1\CaptionSeparator#2%
    \LWR@traceinfo{@makecaption: done}%
}%
}
```

70.4 Caption and LOF linking and tracking

When a new HTML file is marked in the LaTeX PDF file, the LaTeX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special LaTeX labels). This page number is used to generate an `autopage` HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML `autoid` `<id>` at the start of the float itself in the HTML file. The `autopage` and `autoid` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctr `LWR@nextautoid` Tracks `autoid` for floats. Tracks `autopage` for floats.

Ctr `LWR@nextautopage` These are updated per float as the `.lof`, `.lot` file is read.

```
\newcounter{LWR@nextautoid}
\newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat` {⟨*autopage*⟩} {⟨*autoid*⟩}

This is written to the `.lof`, `.lot` file just before each float's usual entry. The `autopage` and `autoid` are remembered for `\l@figure` to use when creating the HTML links.

```
\newcommand*{\LWRsetnextfloat}[2]{%
\setcounter{LWR@nextautopage}{#1}%
\setcounter{LWR@nextautoid}{#2}%
}
```

Ctr `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the `autoid` by the modified `\addcontentsline` just before each float's entry.

```
\newcounter{LWR@latestautopage}
\setcounter{LWR@latestautopage}{1}
```

Env `LWR@figcaption` Encapsulates a caption inside `<figcaption>`, and if `FormatWP` then also a `<div>` with an italic style.

```
\newenvironment*{LWR@figcaption}
{%
\LWR@traceinfo{LWR@figcaption env start}%
\LWR@htmlblocktag{figcaption}%
\ifbool{FormatWP}{%
\begin{BlockClass}[font-style:italic]{italic}
\LWR@print@vspace*{\baselineskip}
}{}%
\LWR@traceinfo{LWR@figcaption env start: done}%
```

```
}
{%
\LWR@traceinfo{LWR@figcaption env end}%
\ifbool{FormatWP}{\end{BlockClass}}{}%
\LWR@htmlblocktag{/figcaption}%
\LWR@traceinfo{LWR@figcaption env end: done}%
}
```

`\LWR@HTML@caption@begin` `{`⟨*type*⟩`}`

Low-level patches to create HTML tags for captions.

```
\newcommand*{\LWR@HTML@caption@begin}[1]
{
\LWR@traceinfo{LWR@HTML@caption@begin}%
```

Keep par and minipage changes local:

```
\begingroup%
```

The **caption** code was not allowing the closing `par` tag:

```
\@setpar{\LWR@closeparagraph\@@par}%
```

No need for a `minipage` or `\parbox` inside the caption:

```
\RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{}%
\RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
\LWR@figcaption%
\LWR@traceinfo{LWR@HTML@caption@begin: about to LWR@origcaption@begin}%
\LWR@print@caption@begin{#1}%
\LWR@traceinfo{LWR@HTML@caption@begin: done}%
}
```

`\LWR@HTML@caption@end` Low-level patches to create HTML tags for captions.

```
\newcommand*{\LWR@HTML@caption@end}
{%
\LWR@traceinfo{LWR@HTML@caption@end}%
\LWR@print@caption@end%
```

Closing tag:

```
\endLWR@figcaption%
```

```
\endgroup%
% \leavevmode% avoid bad space factor (0) error
\LWR@traceinfo{LWR@HTML@caption@end: done}%
}
```

\caption@begin
\caption@end

Low-level patches to create HTML tags for captions. These are assigned \AtBeginDocument so that other packages which modify captions will have already been loaded before saving the print-mode version.

```
\AtBeginDocument{
\LWR@formatted{caption@begin}
\LWR@formatted{caption@end}
}
```

\captionlistentry

Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
\let\LWR@origcaptionlistentry\captionlistentry

\renewcommand*{\captionlistentry}{%
\LWR@ensuredoingapar%
\LWR@origcaptionlistentry%
}

\def\LWR@LTcaptionlistentry{%
\LWR@ensuredoingapar%
\LWR@htmltag{a id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
  \bgroup
  \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
          {\egroup\LWR@LT@captionlistentry}}%
\def\LWR@LT@captionlistentry#1{%
  \caption@listentry\@firstoftwo[\LTcaptype]{#1}}%
```

\addcontentsline

Patched to write the autopage and autoid before each float's entry. No changes if writing .toc For a theorem, automatically defines \ext@<type> as needed, to mimic and reuse the float mechanism.

f

```
\let\LWR@origaddcontentsline\addcontentsline

\renewcommand*{\addcontentsline}[3]{%
\ifstrequal{#1}{toc}{}{% not TOC


\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
    {}%
    {\LWR@newautoidanchor}%
```

```
    \ifcsvoid{ext@#2}{\csdef{ext@#2}{#1}}{}%
    \addtocontents{\@nameuse{ext@#2}}{%
        \protect\LWRsetnextfloat%
        {\arabic{LWR@latestautopage}}%
        {\arabic{LWR@thisautoid}}%
    }%
}% not TOC
\LWR@origaddcontentsline{#1}{#2}{#3}%
}
```

Pkg capt-of
Pkg caption

Either package provides `\captionof`, which is later patched at the beginning of the document.

\captionof

Patched to handle paragraph tags.

```
\AtBeginDocument{
\let\LWR@origcaptionof\captionof

\renewcommand*{\captionof}{%
\LWR@stoppars
\LWR@origcaptionof%
}
}

\end{warpHTML}
```

71 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a `<div>` of class `toc`, `lof`, or `lot`.

A "sidetoc" is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular LaTeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output:

```
\begin{warpHTML}
```

71.1 Reading and printing the TOC

`\LWR@myshorttoc` {⟨*toc/lof/lot/sidetoc*⟩}

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the `@` character into a normal letter to allow formatting commands in the section names.

Unlike in regular LaTeX, the file is not reset after being read, since the sideTOC may be referred to again in each HTML page.

```
\newcommand*{\LWR@myshorttoc}[1]{%
\LWR@traceinfo{LWR@myshorttoc: #1}%
\LWR@ensuredoingapar%
```

Only if the file exists:

```
\IfFileExists{\jobname.#1}{%
\LWR@traceinfo{LWR@myshorttoc: loading}%
```

⚠ Many of the commands in the file will have `@` characters in them, so `@` must be made a regular letter.

```
\begingroup%
\makeatletter%
```

Read in the TOC file:

```
\@input{\jobname.#1}%
\endgroup%
}%
{}%
\LWR@traceinfo{LWR@myshorttoc: done}%
}
```

`\LWR@subtableofcontents` {⟨*toc/lof/lot*⟩} {⟨*sectionstarname*⟩}

Places a TOC/LOF/LOT at the current position.

```
\NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
\@ifundefined{chapter}
{\LWR@closeprevious{\LWR@depthsection}}
{\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
\LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
\@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML `nav` containing the TOC/LOF/LOT:

```
\LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
\LWR@myshorttoc{#1}
```

Close the `nav`:

```
\LWR@htmlelementclassend{nav}{#1}
}
```

`\@starttoc` `{`⟨*ext*⟩`}`

Patch `\@starttoc` to encapsulate the TOC inside HTML tags:

```
\let\LWR@orig@starttoc\@starttoc

\renewcommand{\@starttoc}[1]{
\LWR@htmlelementclass{nav}{#1}
\LWR@orig@starttoc{#1}
\LWR@htmlelementclassend{nav}{#1}
}
```

Bool `LWR@copiedsidetoc` Used to only copy the toc file to the sidetoc a single time.

(**listings** and perhaps other packages would re-use `\tableofcontents` for their own purposes, causing the sidetoc to be copied more than once, and thus end up empty.)

```
\newbool{LWR@copiedsidetoc}
\boolfalse{LWR@copiedsidetoc}
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. **newfloat** uses `\listoffigures` for all future float types.

```
\AtBeginDocument{
```

```
\let\LWR@origtableofcontents\tableofcontents

\renewcommand*{\tableofcontents}{%
```

Do not print the table of contents if formatting for a word processor, which will presumably auto-generate its own updated table of contents:

```
\ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{

=== table of contents ===

}
{
```

Copy the `.toc` file to `.sidetoc` for printing the sideTOC. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
    \ifbool{LWR@copiedsidetoc}{}{%
        \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
        \booltrue{LWR@copiedsidetoc}%
    }%
    \LWR@printpendingfootnotes
    \LWR@origtableofcontents
}
}% \tableofcontents
}% AtBeginDocument
```

\listoffigures

```
\let\LWR@origlistoffigures\listoffigures

\renewcommand*{\listoffigures}{
\ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{

=== list of figures ===

}
{
    \LWR@printpendingfootnotes
    \LWR@origlistoffigures
}
}
```

\listoftables

```
\let\LWR@origlistoftables\listoftables

\renewcommand*{\listoftables}{
```

```
\ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
=== list of tables ===
}
{
    \LWR@printpendingfootnotes
    \LWR@origlistoftables
}
}
```

71.2 High-level TOC commands

`\listof` {⟨*type*⟩} {⟨*title*⟩}

Emulate the `\listof` command from the **float** package (section 201). Used to create lists of custom float types. Also used to redefine the standard LaTeX `\listoffigures` and `\listoftables` commands.

```
\NewDocumentCommand{\listof}{m +m}{%
\@ifundefined{l@#1}{%
    \csdef{l@#1}[2]{\hypertocfloat{1}{#1}{\@nameuse{ext@#1}}{##1}{##2}}%
}{}%
\LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
\expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
\immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
    \jobname.\@nameuse{ext@#1}\relax
}
```

71.3 Side TOC

The "side TOC" is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

CSS may be used to format the sideTOC:

CSS related to sideTOC:

`nav.sidetoc`: The entire sideTOC.

`div.sidetoctitle`: The title.

`div.sidetoccontents`: The table of contents.

```
\end{warpHTML}
```

for HTML & PRINT:

```
\begin{warpall}
```

Ctr `SideTOCDepth` Controls how deep the side-TOC gets. Use a standard LaTeX section level similar to `tocdepth`.

```
\newcounter{SideTOCDepth}
\setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideTOC.

```
\newcommand{\sidetocname}{Contents}
```

```
\end{warpall}
```

for HTML output:

```
\begin{warpHTML}
```

`\LWR@sidetoc` Creates the actual side-TOC.

```
\newcommand*{\LWR@sidetoc}{
\LWR@forcenewpage
\LWR@stoppars

```

The entire sideTOC is placed into a `nav` of class `sidetoc`.

```
\LWR@htmlelementclass{nav}{sidetoc}

\setcounter{tocdepth}{\value{SideTOCDepth}}

```

The title is placed into a `<div>` of class `sidetoctitle`, and may contain paragraphs.

```
\begin{BlockClass}{sidetoctitle}
\sidetocname
\end{BlockClass}
```

The table of contents is placed into a `<div>` of class `sidetoccontents`.

```
\begin{BlockClass}{sidetoccontents}
\LinkHome

\LWR@myshorttoc{sidetoc}
\end{BlockClass}
\LWR@htmlelementclassend{nav}{sidetoc}
}
```

71.4 Low-level TOC line formatting

\numberline {⟨*number*⟩}

(Called from each line in the `.aux`, `.lof` files.)

Record this section number for further use:

```
\newcommand*{\LWR@numberline}[1]{%
\LWR@sectionnumber{#1}\quad%
}

\LetLtxMacro\numberline\LWR@numberline
```

\hypertoc {⟨*1: depth*⟩} {⟨*2: type*⟩} {⟨*3: name*⟩} {⟨*4: page*⟩}

Called by `\l@section`, etc. to create a hyperlink to a section.

The `autopage` label is always created just after the section opens.

#1 is depth

#2 is `section`, `subsection`, etc.

#3 the text of the caption

#4 page number

```
\NewDocumentCommand{\hypertoc}{m m +m m}{%
\LWR@traceinfo{hypertoc !#1!#2!#3!#4!}%
```

Respond to tocdepth:

```
\ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
    \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
    \LWR@subhyperrefclass{%
        \LWR@htmlrefsectionfilename{autopage-#4}\LWR@origpound\LWR@print@mbox{autosec-#4}%
    }{#3}{toc#2}%
    \LWR@stoppars%
}%
{}%
\LWR@traceinfo{hypertoc done}%
}
```

Ctr `lofdepth` TOC depth for figures.

```
\@ifclassloaded{memoir}{}{
\newcounter{lofdepth}
\setcounter{lofdepth}{1}
}
```

Ctr `lotdepth` TOC depth for tables.

```
\@ifclassloaded{memoir}{}{
\newcounter{lotdepth}
\setcounter{lotdepth}{1}
}
```

`\hypertocfloat` {⟨*1: depth*⟩} {⟨*2: type*⟩} {⟨*3: ext of parent*⟩} {⟨*4: caption*⟩} {⟨*5: page*⟩}

#1 is depth

#2 is `figure`, `table`, etc.

#3 is `lof`, `lot`, of the parent.

#4 the text of the caption

#5 page number

```
\newcommand{\hypertocfloat}[5]{%
\LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
\@ifundefined{c@#3depth}{%
\newcounter{#3depth}%
\setcounter{#3depth}{1}%
}{}%
```

Respond to `lofdepth`, etc.:

```
\LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
\ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%
    \LWR@startpars%
```

Create an HTML link to filename#autoid-(float number), with text of the caption, of the given HTML class.

```
    \LWR@subhyperrefclass{%
    \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
    \LWR@origpound\LWR@print@mbox{autoid-\arabic{LWR@nextautoid}}}%
    {#4}{toc#2}%
    \LWR@stoppars%
}{}%
}
```

Automatically called by \contentsline:

\l@part {⟨*name*⟩} {⟨*page*⟩}

Uses \DeclareDocumentCommand in case the class does not happen to have a \part.

```
\DeclareDocumentCommand{\l@part}{m m}{\hypertoc{-1}{part}{#1}{#2}}
```

\l@chapter {⟨*name*⟩} {⟨*page*⟩}

Uses \DeclareDocumentCommand in case the class does not happen to have a \chapter.

```
\DeclareDocumentCommand{\l@chapter}{m m}
    {\hypertoc{0}{chapter}{#1}{#2}}
```

\l@section {⟨*name*⟩} {⟨*page*⟩}

```
\renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
```

\l@subsection {⟨*name*⟩} {⟨*page*⟩}

```
\renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
```

\l@subsubsection {⟨*name*⟩} {⟨*page*⟩}

```
\renewcommand{\l@subsubsection}[2]{\hypertoc{3}{subsubsection}{#1}{#2}}
```

\l@paragraph {⟨*name*⟩} {⟨*page*⟩}

```
\renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
```

\l@subparagraph {⟨*name*⟩} {⟨*page*⟩}

```
\renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
```

`\l@figure` {⟨*name*⟩} {⟨*page*⟩}

```
\renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
```

`\l@table` {⟨*name*⟩} {⟨*page*⟩}

```
\renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

\end{warpHTML}
```

72 Index and glossary

See:
http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx

Index links are tracked by the counter `LWR@autoindex`. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output:

```
\begin{warpHTML}

\newcounter{LWR@autoindex}
\setcounter{LWR@autoindex}{0}

\newcounter{LWR@autoglossary}
\setcounter{LWR@autoglossary}{0}
```

Env `theindex`

```
\@ifundefined{chapter}
{\newcommand*{\LWR@indexsection}[1]{\section*{#1}}}
{\newcommand*{\LWR@indexsection}[1]{\chapter*{#1}}}

\AtBeginDocument{
\renewenvironment*{theindex}{%
    \LWR@indexsection{\indexname}%
    \let\item\LWR@indexitem%
    \let\subitem\LWR@indexsubitem%
    \let\subsubitem\LWR@indexsubsubitem%
}{}
}% AtBeginDocument
```

`\LWR@indexitem` `[`⟨*index key*⟩`]` The optional argument is added to support **repeatindex**.

```
\newcommand{\LWR@indexitem}[1][\@empty]{
\InlineClass{indexitem}{}#1%
}
```

`\LWR@indexsubitem`

```
\newcommand{\LWR@indexsubitem}{
\InlineClass{indexsubitem}{}
}
```

`\LWR@indexsubsubitem`

```
\newcommand{\LWR@indexsubsubitem}{
\InlineClass{indexsubsubitem}{}
}
```

`\@wrindex` `{`⟨*term*⟩`}` Redefined to write the `LWR@autoindex` counter instead of `page`.

```
\def\LWR@wrindex#1{%
\addtocounter{LWR@autoindex}{1}%
\LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
\protected@write\@indexfile{}%
{\string\indexentry{#1}{\arabic{LWR@autoindex}}}%
\endgroup
\@esphack}

\AtBeginDocument{
\let\@wrindex\LWR@wrindex
}
```

`\@wrglossary` `{`⟨*term*⟩`}` Redefined to write the `LWR@latestautopage` counter instead of `page`.

```
\def\@wrglossary#1{%
\addtocounter{LWR@autoglossary}{1}%
\LWR@new@label{LWRglossary-\theLWR@autoglossary}%
\protected@write\@glossaryfile{}%
{\string\glossaryentry{#1}{\theLWR@autoglossary}}%
\endgroup
\@esphack}
```

`\LWR@indexnameref` `{`⟨*LWR@autoindex*⟩`}`

Creates a hyperlink based on the given entry's autoindex.

```
\newcommand*{\LWR@indexnameref}[1]{\nameref{LWRindex-#1}}
```

`\LWR@doindexentry` {⟨*LWR@autoindex, or macros.*⟩}

Creates a hyperlink, or handles `\see`, `\textbf`, etc.

```
\newrobustcmd{\LWR@doindexentry}[1]{%
\IfInteger{#1}%
    {\LWR@indexnameref{#1}}%
    {#1}%
}
```

`\LWR@hyperindexrefnullified` Handles macros commonly seen inside an `\index` entry.

To handle additional macros:

```
\appto\LWR@hyperindexrefnullified{...}
```

```
\newcommand{\LWR@hyperindexrefnullified}{%
\renewcommand{\emph}[1]{\LWR@HTMLemph{\LWR@doindexentry{##1}}}%
\renewcommand{\textbf}[1]{\LWR@HTMLtextbf{\LWR@doindexentry{##1}}}%
\renewcommand{\textrm}[1]{\LWR@HTMLtextrm{\LWR@doindexentry{##1}}}%
\renewcommand{\textsf}[1]{\LWR@HTMLtextsf{\LWR@doindexentry{##1}}}%
\renewcommand{\texttt}[1]{\LWR@HTMLtexttt{\LWR@doindexentry{##1}}}%
\renewcommand{\textup}[1]{\LWR@HTMLtextup{\LWR@doindexentry{##1}}}%
\renewcommand{\textsc}[1]{\LWR@HTMLtextsc{\LWR@doindexentry{##1}}}%
\renewcommand{\textit}[1]{\LWR@HTMLtextit{\LWR@doindexentry{##1}}}%
\renewcommand{\textsl}[1]{\LWR@HTMLtextsl{\LWR@doindexentry{##1}}}%
}
```

`\hyperindexref` {⟨*LWR@autoindex*⟩}

`\hyperindexref{LWR@autoindex}` is inserted into `*.ind` by the **makeindex** style file `lwarp.ist` or the **xindy** style file `lwarp.xdy`.

```
\newcommand{\hyperindexref}[1]{%
\IfInteger{#1}%
    {\LWR@indexnameref{#1}}%
    {%
        \begingroup%
        \LWR@hyperindexrefnullified
        #1%
        \endgroup%
    }%
}
```

```
\end{warpHTML}
```

for PRINT output: A null command for print mode, in case **hyperref** was not used:

```
\begin{warpprint}
\newcommand{\hyperindexref}[1]{#1}
\end{warpprint}
```

for HTML & PRINT: For the **glossaries** package, try to prevent an error where `\glo@name` was not found:

```
\begin{warpall}
\providecommand{\glo@name}{}
\end{warpall}
```

73 Bibliography presentation

for HTML output:

```
\begin{warpHTML}
```

`\bibliography` {⟨*filenames*⟩}

Modified to use the base jobname instead of the `_html` jobname.

```
\def\bibliography#1{%
    \if@filesw
    \immediate\write\@auxout{\string\bibdata{#1}}%
    \fi
%     \@input@{\jobname.bbl}% original
    \begingroup%
    \@input@{\BaseJobname.bbl}% lwarp
    \endgroup%
}
```

`\@biblabel` {⟨*text-refnumber*⟩}

```
\renewcommand{\@biblabel}[1]{[#1]\quad}
```

Env `thebibliography` To emphasize document titles in the bibliography, the following redefines `\em` inside `thebibliography` to gather everything until the next closing brace, then display these tokens with `\textit`.

Adapted from `embracedef.sty`, which is by Takayuki YATO*:*
https://gist.github.com/zr-tex8r/b72555e3e7ad2f0a37f1

```
\AtBeginDocument{
\AtBeginEnvironment{thebibliography}{
\providecommand*{\LWR@newem}[1]{\textit{#1}}

\renewrobustcmd{\em}{%
  \begingroup
    \gdef\LWR@em@after{\LWR@em@finish\LWR@newem}%
    \afterassignment\LWR@em@after
    \toks@\bgroup
}

\def\LWR@em@finish#1{%
    \xdef\LWR@em@after{\noexpand#1{\the\toks@}}%
  \endgroup
  \LWR@em@after\egroup
}
}% \AtBeginEnvironment{thebibliography}
}% \AtBeginDocument

\end{warpHTML}
```

74 Restoring original formatting

\LWR@restoreorigformatting

Used to temporarily restore the print-mode meaning of a number of formatting, graphics, and symbols-related macros while generating SVG math or a `lateximage`.

Must be used inside a group.

Sets `\LWR@formatting` to `print` until the end of the group.

A number of packages will `\appto` additional actions to this macro.

Various packages add to this macro using `\appto`.

for HTML output:

```
\begin{warpHTML}

\newcommand*{\LWR@restoreorigformatting}{%
\LWR@traceinfo{LWR@restoreorigformatting}%
```

Numerous macros change their print/HTML meaning depending on `\LWR@formatting`:

```
\renewcommand*{\LWR@formatting}{print}%
\linespread{1}%

\let\par\LWR@origpar%
```

```
\LWR@select@print@hspace%


\LetLtxMacro\hfil\LWR@orighfil%
\let\hss\LWR@orighss%
\let\llap\LWR@origllap%
\let\rlap\LWR@origrlap%
\let\hfilneg\LWR@orighfilneg%

\let\,\LWR@origcomma% disable HTML short unbreakable space
\let\thinspace\LWR@origthinspace% disable HTML short unbreakable space
\let\textellipsis\LWR@origtextellipsis%
\let\textless\LWR@origtextless%
\let\textgreater\LWR@origtextgreater%
\LetLtxMacro\textrm\LWR@origtextrm%
\LetLtxMacro\textsf\LWR@origtextsf%
\LetLtxMacro\texttt\LWR@origtexttt%
\LetLtxMacro\textbf\LWR@origtextbf%
\LetLtxMacro\textmd\LWR@origtextmd%
\LetLtxMacro\textit\LWR@origtextit%
\LetLtxMacro\textsl\LWR@origtextsl%
\LetLtxMacro\textsc\LWR@origtextsc%
\LetLtxMacro\textup\LWR@origtextup%
\LetLtxMacro\textnormal\LWR@origtextnormal%
\LetLtxMacro\emph\LWR@origemph%
\LetLtxMacro\rmfamily\LWR@origrmfamily%
\LetLtxMacro\sffamily\LWR@origsffamily%
\LetLtxMacro\ttfamily\LWR@origttfamily%
\LetLtxMacro\bfseries\LWR@origbfseries%
\LetLtxMacro\mdseries\LWR@origmdseries%
\LetLtxMacro\upshape\LWR@origupshape%
\LetLtxMacro\slshape\LWR@origslshape%
\LetLtxMacro\scshape\LWR@origscshape%
\LetLtxMacro\itshape\LWR@origitshape%
\LetLtxMacro\em\LWR@origem%
\LetLtxMacro\normalfont\LWR@orignormalfont%
\let\sp\LWR@origsp%
\let\sb\LWR@origsb%
\LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
\LetLtxMacro\@textsuperscript\LWR@orig@textsuperscript%
\LetLtxMacro\textsubscript\LWR@origtextsubscript%
\LetLtxMacro\@textsubscript\LWR@orig@textsubscript%
\LetLtxMacro\underline\LWR@origunderline%
\let~\LWR@origtilde%
\let\enskip\LWR@origenskip%
\let\quad\LWR@origquad%
\let\qquad\LWR@origqquad%
\LetLtxMacro\tabular\LWR@origtabular%
\LetLtxMacro\endtabular\LWR@origendtabular%
\LetLtxMacro\noalign\LWR@orignoalign%
```

```
\LetLtxMacro\hline\LWR@orighline%
\let\newline\LWR@orignewline%
\LetLtxMacro\includegraphics\LWR@origincludegraphics%
\let\TeX\LWR@origTeX%
\let\LaTeX\LWR@origLaTeX%
\let\LaTeXe\LWR@origLaTeXe%
\renewcommand*{\Xe}{X\textsubscript{E}}%

\LetLtxMacro\@ensuredmath\LWR@origensuredmath%
%
\LWR@restoreorigaccents%
\LWR@restoreoriglists%
%
\LWR@FBcancel%
}

\end{warpHTML}
```

75 Math

75.1 Limitations

75.1.1 Rendering tradeoffs

Math rendering — Math may be rendered as SVG graphics or using the MathJax JavaScript display engine.

SVG files — Rendering math as images creates a new SVG file for each expression, except that an MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to SVG only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change.

SVG inline — The SVG images are currently stored separately, but they could be encoded in-line directly into the HTML document. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files — Others LaTeX-to-HTML converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.

MathML — Conversion to MathML might be a better approach, among other things allowing a

more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 10 regarding EPUB output with MathJax.

75.1.2 SVG option

SVG math option

For SVG math, math is rendered as usual by LaTeX into the initial PDF file using the current font[17], then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the LaTeX code which generated the math, allowing copy/paste of the LaTeX math expression into other documents.

SVG image font size

For the `lateximage` environment, the size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, which defaults to:

```
\renewcommand{\LateximageFontSizeName}{normalsize}
```

For inline SVG math, font size is instead controlled by `\LateximageFontScale`, which defaults to:

```
\newcommand*{\LateximageFontScale}{.75}
```

SVG math copy/paste

For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the LaTeX code for the math expression. For $\mathcal{AMS}$ environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred $\mathcal{AMS}$ environment, although the `\tag` macro will still appear inside the LaTeX math expression.

⚠ SVG math in TeX boxes

SVG math does not work inside TeX boxes, since a `\newpage` is required before and after each image.

75.1.3 MathJax option

MathJax math option

Prog MathJax

The popular MathJax alternative (mathjax.org) may be used to display math.

When MathJax is enabled, math is rendered twice:

1. As regular LaTeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of LaTeX, and

[17] See section 411 regarding fonts and fractions.

2. As detokenized printed LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current LaTeX values, and the MathJax cross-referencing system is ignored in favor of the LaTeX internal system, seamlessly integrating with the rest of the LaTeX code.

75.1.4 Customizing MathJax

MathJax does not have preexisting support every possible math function. Additional MathJax function definitions may be defined. These will be declared at the start of each HTML page, and thus will have a global effect.

Examples:

```
\CustomizeMathJax{
    \newcommand{\expval}[1]{\langle#1\rangle}
    \newcommand{\abs}[1]{\lvert#1\rvert}
}
\CustomizeMathJax{\newcommand{\arsinh}{\text{arsinh}}}
\CustomizeMathJax{\newcommand{\arcosh}{\text{arcosh}}}
\CustomizeMathJax{\newcommand{\NN}{\mathbb{N}}}
```

75.1.5 MathJax limitations

MathJax limitations

Prog MathJax

Limitations when using MathJax include:

chapter numbers

- In document classes which have chapters, `\tagged` equations have the chapter number prepended in HTML output, unlike LaTeX. `\tag*` equations (correctly) do not. This may be improved with future versions of the MathJax support script.

  https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY

subequations

- MathJax itself does not support subequations. This may be improved by parsing the LaTeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

- Footnotes inside equations are not yet supported while using MathJax.

lateximage

- Math appearing inside a lateximage, and therefore also inside a Ti*k*z or `picture` environment, is rendered as SVG math even if MathJax is used in the rest of the document.

siunitx

- Usage of **siunitx** inside a math equation is supported via a third-party MathJax

⚠ siunitx inside an equation

extension. While inside a math expression, do not use \SI or \si inside \text, where it will be rendered as normal text.

https://github.com/burnpanck/MathJax-siunitx

Also see section 9.6.12.

tabbing

- A tabbing environment is emulated using an HTML <pre>.While MathJax is enabled inside tabbing, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.

⚠ other macros and packages

- Other math-related macros and packages are not supported by MathJax, including \ensuremath, **bigdelim**, **units**, and **nicefrac**, along with occasionally-used macros such as \footnote and \relax.

75.1.6 Catcode changes

preamble macros with math

The math shift character $ is not set for HTML output until after the preamble. Macros defined in the preamble which contain $ must be enclosed between \StartDefiningMath and \StopDefiningMath to temporarily change to the HTML meaning of $:

```
\StartDefiningMath
\newcommand{...}
\StopDefiningMath
```

As an alternative, use \(and \) instead of $, in which case \StartDefiningMath and \StopDefiningMath are not necessary.

If a package defines macros using $, it may be nessary to use \StartDefiningMath and \StopDefiningMath before and after loading the package.

75.1.7 Dynamic math

inline dynamic math

An inline math expression is usually converted to a reusable hashed SVG math image, or a MathJax expression. The hash or expression depends on the contents of the math expression. In most cases this math expression is static, such as $x+1$, so the image can be reused for multiples instances of the same expression. In some cases, the math expression includes a counter or other object which may change between uses. The macro \StartDynamicMath may be used before a dynamic math expression, and \StopDynamicMath after. Doing so tells **lwarp** to use an unhashed SVG math image, even if MathJax is in use. See section 41.

75.1.8 Display math

\displaymathnormal By default, or when selecting \displaymathnormal, math display environments print their contents in MathJax, and render their contents in svg math as well as use their contents in the alt tag of html output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated Ti*k*z pictures, compilation will fail.

\displaymathother When selecting \displaymathother, it is assumed that the contents are more complicated than "pure" math. An example is an elaborate Ti*k*z picture, which will not render in MathJax and will not make sense as an html alt tag. In this mode, MathJax is turned off, math display environments become svg images, even if MathJax is selected, and the html alt tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as Ti*k*z pictures are more likely to compile successfully.

75.2 HTML alt tag names

Redefinable names for the html alt tags, for translation according to the reader's native language.

for HTML & PRINT:

```
\begin{warpall}
```

\mathimagename Default: "math image"

The html alt tag for an svg math image.

```
\newcommand*{\mathimagename}{math image}
```

\packagediagramname Default: "diagram"

Appended to the lateximage html alt tag for the images generated by many packages.

```
\newcommand*{\packagediagramname}{diagram}
```

```
\end{warpall}
```

75.3 Inline and display math

for HTML output:

```
\begin{warpHTML}
```

Ctr LWR@externalfilecnt

Counter for the external files which are generated and then referenced from the html:

```
\newcounter{LWR@externalfilecnt}
```

Bool `LWR@indisplaymathimage` True if processing display math for SVG output. Inside a `lateximage`, display math is only set to print-mode output if `LWR@indisplaymathimage` is false. Used to avoid nullifying display math before it has been completed.

```
\newbool{LWR@indisplaymathimage}
```

`\$` Plain dollar signs appearing in the HTML output may be interpreted by MathJax to be math shifts. For a plain text dollar `\$`, use an HTML entity to avoid it being interpreted by MathJax, unless are inside a lateximage, in which case it will not be seen by MathJax.

```
\let\LWR@origtextdollar\$

\renewcommand*{\$}{%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{\LWR@origtextdollar}%
{\HTMLentity{dollar}}%
}
```

File `lwarp_baseline_marker.png` File `lwarp_baseline_marker.eps` A marker to be used to help **pdfcrop** identify the inline math baseline and width. If either **graphicx** or **graphics** is loaded, this marker is placed at the lower left and lower right corners of the inline math. **pdfcrop** is then able to identify the width of the image, and also the height of an image such as a horizontal dash which does not otherwise touch the baseline.

A marker with alpha or opacity of 0% is not registered by **pdfcrop**, so the marker is a small square block of 1% alpha, which seems to work while still being effectively invisible in the final SVG image.

If **graphicx** is loaded, this marker is sized as a tiny 1 sp square. If **graphics** is loaded, this marker is used at its default size of around .25 pt. If neither graphics package is loaded, the marker is replaced by a 10 sp horizontal space, and there is no assistance for determining baseline or width of the inline math image. The best results are obtained when using **graphicx**.

`\LWR@addbaselinemarker` Places a small marker in an SVG inline image. If **graphics** or **graphicx** are loaded, the marker is a mostly transparent image. If neither is loaded, no marker is used.

```
\AtBeginDocument{

\ifpdf
    \newcommand*{\LWR@baselinename}{lwarp_baseline_marker.png}
\else
    \ifXeTeX
        \newcommand*{\LWR@baselinename}{lwarp_baseline_marker.png}
    \else
        \newcommand*{\LWR@baselinename}{lwarp_baseline_marker.eps}
```

```
    \fi
\fi

\IfFileExists{\LWR@baselinename}%
{
    \@ifpackageloaded{graphicx}{
        \newcommand*{\LWR@addbaselinemarker}{%
            \LWR@origincludegraphics{\LWR@baselinename}%
        }
    }{
        \@ifpackageloaded{graphics}{
            \newcommand*{\LWR@addbaselinemarker}{%
                \LWR@origincludegraphics{\LWR@baselinename}%
            }
        }{
            \PackageWarning{lwarp}{Load graphicx or graphics
                for improved SVG math baselines,}
            \newcommand*{\LWR@addbaselinemarker}{}
        }
    }
}{% lwarp_baseline_marker.png or .eps is not present
    \PackageWarning{lwarp}{File \LWR@baselinename\space is not installed alongside
        the lwarp-*.sty files, so SVG math baselines may not be accurate,}
    \newcommand*{\LWR@addbaselinemarker}{}
}

}% AtBeginDocument
```

`\LWR@subsingledollar` * {⟨*2: alt text*⟩} {⟨*3: add'l hashing*⟩} {⟨*4: math expression*⟩}

For inline math. Uses MathJax, or for SVG math the image is measured and ajusted to the baseline of the HTML output, and placed inside a `lateximage`.

image filename hashing

If starred, a hashed filename is used. If so, the hash is based on the `alt` tag and also the additional hashing argument.

This may be used to provide an expression with a simple `alt` tag but also enough additional information to provide a unique hash.

An example is when the expression is a complicated TEX expression, which would not copy/paste well. A simplified tag may be used, while the complicated expression is duplicated in the additional hashing argument.

Another example is when the expression is simple, but the image depends on options. These options may be decoded into text form and included in the additional hashing argument in order to make the hash unique according to the set of options, even if the simple `alt` tag is still the same.

```
\newlength{\LWR@singledollarwidth}
```

```
\newlength{\LWR@singledollarheight}
\newlength{\LWR@singledollardepth}

\newsavebox{\LWR@singledollarbox}

\NewDocumentCommand{\LWR@subsingledollar}{s m m m}{%
\LWR@traceinfo{LWR@subsingledollar}%

\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
{%
\LWR@traceinfo{LWR@subsingledollar: already in a lateximage}%
    #4% contents
}%
{% not in a lateximage
\begingroup%
```

MathJax cannot parse the often complicated TEX expressions which appear in the various uses of `\ensuredmath`. `\ensuremath` forces the `alt` tag to "`(math image)`", as translated according to `\mathimagename`. If this is the case, force the use of a `lateximage` even if MathJax. Likewise for **siunitx** if `parse-numbers=false`.

If MathJax, or if formatting math for a word processor, and not `\ensuredmath`, and not a dynamic math expression, print the math expression:

```
\ifboolexpr{%
    (
        bool{mathjax} or
        ( bool{FormatWP} and bool{WPMarkMath} )
    ) and
    ( not test { \ifstrequal {#2} {(\mathimagename)} } ) and % from \ensuredmath
    ( not bool{LWR@dynamicmath} )
}%
```

For MATHJAX, print the math between `\(` and `\)`:

```
{%
    \LWR@traceinfo{LWR@subsingledollar: Mathjax}%
    {\textbackslash(\LWR@HTMLsanitize{#4}\textbackslash)}%
}% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the LATEX code, and a CSS style to control the baseline adjustment.

```
{% not mathjax
\LWR@traceinfo{LWR@subsingledollar: NOT mathjax, or is ensuremath, or is dynamic}%
```

Measure the depth, width, and height of the math image:

```
    \begingroup%
```

Temporarily disable formatting while measuring the image parameters:

```
    \LWR@restoreorigformatting%
    \RenewDocumentEnvironment{lateximage}{s o o o}{}{}% inside group
    \LWR@print@normalsize%
```

Temporarily set font for the HTML PDF output:

```
    \LWR@traceinfo{Using font family \LWR@f@family}%
    \@nameuse{LWR@orig\LWR@f@family family}%
    \LWR@traceinfo{Using font series \LWR@f@series}%
    \@nameuse{LWR@orig\LWR@f@series series}%
    \LWR@traceinfo{Using font shape \LWR@f@shape}%
    \@nameuse{LWR@orig\LWR@f@shape shape}%
```

`lateximagedepth` must be nested to avoid generating paragraph tags. $\mathcal{AMS}$ math modifies the `\text` macro such that `\addtocounter` does not always occur as expected. Lower-level code is used instead.

```
    \global\advance\c@LWR@lateximagedepth 1\relax%
```

Typeset and save the contents, depending on how they were generated:

SVG math: `\LWR@origensuredmath` is part of argument #4.

SVG math `\ensuremath`: `\LWR@origensuredmath` is part of argument #4.

SVG dynamic math: `\LWR@origensuredmath` is part of argument #4.

Mathjax: Argument #4 is the contents of the math expression without `\LWR@origensuredmath`. This case is handled above.

Mathjax `\ensuremath`: `\LWR@origensuredmath` is part of argument #4.

Mathjax dynamic math: Argument #4 is the contents of the math expression without `\LWR@origensuredmath`, so `\LWR@origensuredmath` is added below.

`\ifmmode`: Included "just in case".

```
    \ifmmode%
        \global\sbox{\LWR@singledollarbox}{#4}%
    \else%
        \ifbool{LWR@dynamicmath}{%
            \ifbool{mathjax}{%
                \global\sbox{\LWR@singledollarbox}{\LWR@origensuredmath{#4}}%
            }{%
```

```
                \global\sbox{\LWR@singledollarbox}{#4}%
            }%
        }{%
            \global\sbox{\LWR@singledollarbox}{#4}%
        }%
    \fi%
```

Add a small and almost transparent marker at the depth of the image.

A math minus sign has the same depth as a plus, even though it does not draw anything below the baseline. This means that **pdfcrop** would crop the image without depth. The marker below the baseline is seen by **pdfcrop** and preserves the depth.

```
    \global\sbox{\LWR@singledollarbox}{%
        \usebox{\LWR@singledollarbox}%
        \raisebox{-\dp\LWR@singledollarbox}{%
            \LWR@addbaselinemarker%
        }%
    }%
```

More low-level code to undo the counter change.

```
    \global\advance\c@LWR@lateximagedepth -1\relax% Due to AmS \text macro.
```

Measure the depth:

```
    \setlength{\LWR@singledollardepth}{%
        \LateximageFontScale\dp\LWR@singledollarbox%
    }%
```

Make the length a global change:

```
    \global\LWR@singledollardepth=\LWR@singledollardepth%
```

Likewise for width:

```
    \setlength{\LWR@singledollarwidth}{%
        \LateximageFontScale\wd\LWR@singledollarbox%
    }%
    \global\LWR@singledollarwidth=\LWR@singledollarwidth%
```

Likewise for total height:

```
    \setlength{\LWR@singledollarheight}{%
        \LateximageFontScale\ht\LWR@singledollarbox%
    }%
    \addtolength{\LWR@singledollarheight}{%
        \LateximageFontScale\dp\LWR@singledollarbox%
```

```
    }%
    \global\LWR@singledollarheight=\LWR@singledollarheight%

    \endgroup%
```

Set a style for the the height or width. The em unit is used so that the math scales according to the user's selected font size.

Start with the greater of the width or the height, biased towards the width:

```
    \ifdimgreater{\LWR@singledollarwidth}{.7\LWR@singledollarheight}{%
        \def\LWR@singledollarstyle{%
            width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
        }%
    }{%
        \def\LWR@singledollarstyle{%
            height:\LWR@convertto{em}{\the\LWR@singledollarheight} em%
        }%
    }%
```

If a very narrow width, use the height.

```
    \ifdimless{\LWR@singledollarwidth}{.2em}%
    {%
        \def\LWR@singledollarstyle{%
            height:\LWR@convertto{em}{\the\LWR@singledollarheight} em%
        }%
    }%
    {}%
```

If very wide and short, use the width:

```
    \ifdimless{\LWR@singledollarheight}{.2em}%
    {%
        \def\LWR@singledollarstyle{%
            width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
        }%
    }%
    {}%
```

If there is significant text depth, add the depth to the style.

```
    \ifdimgreater{\LWR@singledollardepth}{0.05ex}{%
        \def\LWR@singledollardepthstyle{%
            \ ; % extra space
            \LWR@print@mbox{%
                vertical-align:-\LWR@convertto{em}{\the\LWR@singledollardepth} em%
            } % extra space
```

```
        }%
    }{%
        \def\LWR@singledollardepthstyle{}%
    }%
```

Create the `lateximage` using the alternate tag and the computed size and depth. The star causes `lateximage` to use an MD5 hash as the filename. When hashing, also include the current font and color in the hash.

```
    \ifbool{LWR@dynamicmath}{%
        \LWR@traceinfo{subsingledollar: dynamic}%
        \begin{lateximage}% no hashing
            [(\mathimagename)]% alt tag
            []% no add'l hashing
            [\LWR@singledollarstyle \LWR@singledollardepthstyle]% CSS
    }{%
        \LWR@traceinfo{subsingledollar: static}%
        \IfValueTF{#1}{%
            \LWR@findcurrenttextcolor% sets \LWR@tempcolor
            \begin{lateximage}*% use hashing
                [#2]% alt
                [% addl' hashing
                    #3%
                    FM\LWR@f@family%
                    SR\LWR@f@series%
                    SH\LWR@f@shape%
                    CL\LWR@tempcolor%
                ]%
                [\LWR@singledollarstyle \LWR@singledollardepthstyle]% CSS
        }{%
            \begin{lateximage}% no hashing
                [#2]% alt
                []% no add'l hashing
                [\LWR@singledollarstyle \LWR@singledollardepthstyle]% CSS
        }%
    }%
```

Place small and almost transparent markers on the baseline at the left and right edges of the image. These markers are seen by **pdfcrop**, and force vertically-centered objects such as a dash to be raised off the baseline in the cropped image, and also force the total width and left / right margins to be correct. (Except that in some fonts a character may exceed the bounding box, and thus may appear wider than expected when converted to an image.)

```
    \LWR@addbaselinemarker%
```

Typeset the contents:

```
    \usebox{\LWR@singledollarbox}%
```

The closing baseline marker:

```
    \LWR@addbaselinemarker%

    \end{lateximage}%
%
}% not mathjax
\endgroup%
}% not in a lateximage
\LWR@traceinfo{LWR@subsingledollar: done}%
}

\LetLtxMacro\LWR@origdollar$
\LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

\LetLtxMacro\LWR@origopenparen\(
\LetLtxMacro\LWR@origcloseparen\)
\LetLtxMacro\LWR@origopenbracket\[
\LetLtxMacro\LWR@origclosebracket\]
```

$ $$ Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:

```
\begingroup
\catcode`\$=\active%
\protected\gdef${\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

Used by **chemformula** to escape single-dollar math:

```
\protected\gdef\LWR@newsingledollar{\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

\LWR@doubledollar Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
\protected\gdef\LWR@doubledollar$#1$${%
```

If MathJax or formatting for a word processor, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between `\[` and `\]`:

```
{

    \textbackslash[%
    \LWR@HTMLsanitize{#1}%
    \textbackslash]
```

```
}% mathjax
```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the LaTeX code:

```
{% not mathjax
    \begin{BlockClass}{displaymath}%
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}%
    [%
        \textbackslash{[} % extra space
        \LWR@HTMLsanitize{#1} % extra space
        \textbackslash{]}%
    ]%
    \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
    \end{lateximage}%
    \end{BlockClass}%
}% not mathjax
}%
```

`\LWR@singledollar` {⟨*alt text*⟩} {⟨*math expression*⟩}

```
\protected\gdef\LWR@singledollar#1${%
\ifbool{mathjax}{%
    \LWR@subsingledollar*%
    {% alt tag
        \textbackslash( %
        \LWR@HTMLsanitize{#1} % extra space
        \textbackslash)%
    }%
    {singledollar}% add'l hashing
    {#1}% contents
}{% not mathjax
    \LWR@subsingledollar*%
    {% alt tag
        \textbackslash( %
        \LWR@HTMLsanitize{#1} % extra space
        \textbackslash)%
    }%
    {singledollar}% add'l hashing
    {\LWR@origensuredmath{#1}}% contents
}% not mathjax
}
```

`\(` `\[` Redefine to the above dollar macros.

```
\AtBeginDocument{
```

```
\protected\gdef\(#1\){$#1$}
\protected\gdef\[#1\]{$$#1$$}
}

\endgroup


\AtBeginDocument{
\LetLtxMacro\LWR@openbracketnormal\[
\LetLtxMacro\LWR@closebracketnormal\]
}
```

`\@ensuredmath` {⟨*expression*⟩}

If MathJax, a `lateximage` is used, since `\ensuremath` is often used for complex TEX expressions which MathJax may not render. If SVG math, a hashed file is used with a simple `alt` tag, but additional hashing provided by the contents.

```
\LetLtxMacro\LWR@origensuredmath\@ensuredmath

\renewcommand{\@ensuredmath}[1]{%
\ifbool{mathjax}{%
    \LWR@subsingledollar*{(\mathimagename)}{%
           \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
    }{\relax%
        \LWR@origensuredmath{#1}%
    }%
}{% SVG math
```

If already inside a `lateximage` in math mode, continue as-is.

```
\ifmmode%
    \LWR@origensuredmath{#1}%
\else%
```

Create an inline math `lateximage` with a simple `alt` tag and additional hashing according to the contents.

```
    \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
    {\LWR@origensuredmath{#1}}%
    {%
        \LWR@subsingledollar*{(\mathimagename)}{%
           \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
        }{%
            \LWR@origensuredmath{#1}%
        }%
    }%
\fi%
```

```
}%
}
```

Remove the old `math` and `displaymath` environments:

```
\let\math\relax
\let\endmath\relax
\let\displaymath\relax
\let\enddisplaymath\relax
```

Env `math` Set math mode then typeset the body of what was between the begin/end. See the **environ** package for `\BODY`.

```
\NewEnviron{math}{\expandafter\(\BODY\)}
```

Env `LWR@displaymathnormal` Set math mode then typeset the body of what was between the begin/end. See the **environ** package for `\BODY`.

```
\NewEnviron{LWR@displaymathnormal}{\expandafter\[\BODY\]\@ignoretrue}
```

Set the default displaymath to the normal version:

```
\LetLtxMacro\[\LWR@openbracketnormal%
\LetLtxMacro\]\LWR@closebracketnormal%
\LetLtxMacro\displaymath\LWR@displaymathnormal%
\LetLtxMacro\enddisplaymath\endLWR@displaymathnormal%
```

Env `LWR@displaymathother` A version of `displaymath` which can handle complicated objects, but does not supply MathJax or HTML `alt` tags.

```
\newenvironment{LWR@displaymathother}
{%
    \begin{BlockClass}{displaymath}%
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}%
    [(display math)]%
    \LWR@origdollar\LWR@origdollar%
}
{%
    \LWR@origdollar\LWR@origdollar%
    \end{lateximage}%
    \end{BlockClass}%
}
```

Env LWR@equationother A version of displaymath which can handle complicated objects, but does not supply MathJax or HTML alt tags.

```
\newenvironment{LWR@equationother}
{%
    \begin{BlockClass}{displaymathnumbered}%
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}%
    [(display math)]%
    \LWR@origequation%
}
{%
    \LWR@origendequation%
    \end{lateximage}%
    \end{BlockClass}%
}
```

75.4 MathJax support

Ctr LWR@nextequation Used to add one to compute the next equation number.

```
\newcounter{LWR@nextequation}
```

\LWR@syncmathjax Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside "\(" and "\)" characters. They are printed to HTML output, not interpreted by LaTeX.

```
\newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
\ifcsdef{thechapter}{
\InlineClass{hidden}{
\textbackslash(
\textbackslash{}seteqsection \{\thechapter\}
\textbackslash)
}
}
{}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
\ifthenelse{\cnttest{\value{equation}}>0}
{
```

Tell MathJax that the next set of equations begins with the current LaTeX equation number, plus one.

```
    \setcounter{LWR@nextequation}{\value{equation}}
    \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside "\(" and "\)" characters, to be printed to HTML, not interpreted by LaTeX.

```
    \InlineClass{hidden}{
        \textbackslash(
        \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
        \textbackslash)
    }
}{}% not eq > 0
}
```

`\LWR@hidelatexequation` {⟨*environment*⟩} {⟨*contents*⟩}

Creates the LaTeX version of the equation inside an HTML comment.

```
\NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
\LWR@stoppars
\LWR@htmlopencomment

```

Start the LaTeX math environment inside the HTML comment:

```
\begingroup
\@nameuse{LWR@orig#1}
```

While in the math environment, restore various commands to their LaTeX meanings.

```
\LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 75.7.1.

Print the contents of the equation:

```
#2
```

End the LaTeX math environment inside the HTML comment:

```
\@nameuse{LWR@origend#1}
\endgroup
```

Close the HTML comment and resume HTML paragraph handling:

```
\LWR@htmlclosecomment
\LWR@startpars
}
```

`\LWR@addmathjax` `{`⟨*environment*⟩`}` `{`⟨*contents*⟩`}`

Given the name of a math environment and its contents, create a MathJax instance. The contents are printed to HTML output, not interpreted by LaTeX.

```
\NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed "`\(`" and "`\)`" characters.

```
\LWR@origtilde\LWR@orignewline
\textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
\LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
\textbackslash{}end\{#1\}
\LWR@orignewline
}
```

75.5 Equation environment

Remember existing `equation` environment:

```
\AtBeginDocument{
\let\LWR@origequation\equation
\let\LWR@origendequation\endequation
\csletcs{LWR@origequation*}{equation*}
\csletcs{LWR@origendequation*}{endequation*}
}
```

\LWR@doequation {⟨*env contents*⟩} {⟨*env name*⟩}

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an `<alt>` tag containing a detokenized copy of the LaTeX source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\label`s. The contents are also printed to the HTML output for processing by the MathJax script.

```
\newcommand*{\LWR@doequation}[2]{%
```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

MathJax output:

```
{
```

Print commands to syncronize MathJax's equation number and format to the current LaTeX chapter/section and equation number:

```
    \LWR@syncmathjax
```

Print the LaTeX math inside an HTML comment:

```
    \LWR@hidelatexequation{#2}{#1}
}
```

SVG output: Create the `lateximage` along with an HTML `<alt>` tag having an equation number, the LaTeX equation environment commmands, and the contents of the environment's `\BODY`.

```
{% not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
    \ifstrequal{#2}{equation*}{%
        \begin{BlockClass}{displaymath}%
    }{%
        \begin{BlockClass}{displaymathnumbered}%
    }%
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[%
        \ifstrequal{#2}{equation*}{%
```

```
            \ifdefequal{\LWR@equationtag}{\theequation}{%
%                                       no tag was given
            }{%
                (\LWR@equationtag) % tag was given
            }%
        }{%
            (\LWR@equationtag) % automatic numbering
        }%
        \textbackslash{begin\{#2\}} % extra space
        \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#1}} % extra space
        \textbackslash{end\{#2\}}%
    ]% alt tag
```

Create the actual LaTeX-formatted equation inside the `lateximage` using the contents of the environment.

```
    \@nameuse{LWR@orig#2}%
    #1% contents collected by \collect@body
    \@nameuse{LWR@origend#2}%
    \end{lateximage}%
    \end{BlockClass}%
}% not mathjax
}
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
\newcommand*{\LWR@doendequation}[1]{%
    \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
    {%
        \LWR@addmathjax{#1}{\BODY}%
    }{}%

}
```

Remove existing `equation` environment:

```
\AtBeginDocument{
\let\equation\relax
\let\endequation\relax
\csletcs{equation*}{relax}
\csletcs{endequation*}{relax}
}
```

Env `equation` The new `equation` environment is created with `\NewEnviron` (from the **environ** package), which stores the contents of its environment in a macro called `\BODY`.

```
\AtBeginDocument{
\NewEnviron{equation}%
{\LWR@doequation{\BODY}{equation}}%
[\LWR@doendequation{equation}]

\LetLtxMacro\LWR@equationnormal\equation
\LetLtxMacro\endLWR@equationnormal\endequation
}
```

Env `equation*`

```
\AtBeginDocument{
\NewEnviron{equation*}%
{\LWR@doequation{\BODY}{equation*}}%
[\LWR@doendequation{equation*}]

\csletcs{LWR@equationnormalstar}{equation*}
\csletcs{LWR@endequationnormalstar}{endequation*}
}
```

Remember the "less" version of `equation`, which uses MathJax and `alt` tags, but does not support complicated contents such as some Ti*k*z expressions.

```
\AtBeginDocument{
\LetLtxMacro\LWR@equationless\equation
\LetLtxMacro\endLWR@equationless\endequation
\csletcs{LWR@equationlessstar}{equation*}
\csletcs{LWR@endequationlessstar}{endequation*}
}
```

75.6 `\displaymathnormal` and `\displaymathother`

`\displaymathnormal` By default, or when selecting `\displaymathnormal`, math display environments print their contents in MathJax, and render their contents in SVG math as well as use their contents in the `alt` tag of HTML output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated Ti*k*z pictures, compilation will fail.

`\displaymathother` When selecting `\displaymathother`, it is assumed that the contents are more complicated than "pure" math. An example is an elaborate Ti*k*z picture, which will not render in MathJax and will not make sense as an HTML `alt` tag. In this mode, MathJax is turned off, math display environments become SVG images, even if MathJax is selected, and the HTML `alt` tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as Ti*k*z pictures are more likely to compile successfully.

`\displaymathnormal`
simple math objects

Use when display math environments have simple math which is to sent to MathJax or included in HTML `alt` tags.

```
\newcommand*{\displaymathnormal}{%
\ifbool{LWR@origmathjax}{\booltrue{mathjax}}{\boolfalse{mathjax}}%
\LetLtxMacro\[\LWR@openbracketnormal%
\LetLtxMacro\]\LWR@closebracketnormal%
\LetLtxMacro\displaymath\LWR@displaymathnormal%
\LetLtxMacro\enddisplaymath\endLWR@displaymathnormal%
\LetLtxMacro\equation\LWR@equationnormal%
\LetLtxMacro\endequation\endLWR@equationnormal%
\csletcs{equation*}{LWR@equationnormalstar}%
\csletcs{endequation*}{LWR@endequationnormalstar}%
}
```

`\displaymathother`
complicated math objects

Use when display math environments have complicated objects which will not work with MathJax or should not be included in HTML `alt` tags. Complicated contents are more likely to compile correctly.

```
\newcommand*{\displaymathother}{%
\boolfalse{mathjax}%
\LetLtxMacro\displaymath\LWR@displaymathother%
\LetLtxMacro\enddisplaymath\endLWR@displaymathother%
\LetLtxMacro\[\LWR@displaymathother%
\LetLtxMacro\]\endLWR@displaymathother%
\LetLtxMacro\equation\LWR@equationother%
\LetLtxMacro\endequation\endLWR@equationother%
\csletcs{equation*}{displaymath}%
\csletcs{endequation*}{enddisplaymath}%
}
```

```
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
```

Print-mode versions:

```
\newcommand*{\displaymathnormal}{}
\newcommand*{\displaymathother}{}
```

```
\end{warpprint}
```

for HTML output:

```
\begin{warpHTML}
```

75.7 AMS Math environments

75.7.1 Support macros

Bool `LWR@amsmultline` True if processing a multline environment.

To compensate for `multline`-spefific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an **amsmath** environment which is not a `multline` environment and not an `equation`.

```
\newbool{LWR@amsmultline}
\boolfalse{LWR@amsmultline}
```

`\LWR@htmlmathlabel` {⟨*label*⟩}

lwarp points `\ltx@label` here. This is used by `\label` when inside a LaTeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the LaTeX original, modified by **lwarp**, then by **amsmath**, then by **cleveref**.

```
\newcommand*{\LWR@htmlmathlabel}[1]{%
\LWR@traceinfo{LWR@htmlmathlabelb #1}%
```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{%
```

The combined LaTeX & HTML label is printed in a `\text` field:

```
    \text{%
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
    \ifbool{LWR@amsmultline}{}{\hspace*{\totwidth@}}%
```

Temporarily end the HTML comment, insert the LaTeX & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the **amsmath** package.)

```
    \LWR@htmlclosecomment%
    \LWR@origltx@label{#1}%
    \LWR@htmlopencomment%
    }% text
```

```
}% mathjax
{%
    \LWR@origltx@label{#1}%
}%
}
```

`\LWR@beginhideamsmath` Starts hiding LaTeX math inside an HTML comment.

```
\newcommand*{\LWR@beginhideamsmath}{
\LWR@stoppars
\LWR@origtilde\LWR@orignewline
\LWR@htmlopencomment

\begingroup
\LWR@restoreorigformatting
}
```

`\LWR@endhideamsmath` Ends hiding LaTeX math inside an HTML comment.

```
\newcommand*{\LWR@endhideamsmath}{
\endgroup

\LWR@htmlclosecomment
\LWR@orignewline
\LWR@startpars
}
```

75.7.2 Environment patches

The **amsmath** environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an $\mathcal{AMS}$ package, and thus requires special handling.

For SVG math: Each envrionment is encapsulated inside a `lateximage` environment, along with a special optional argument of `\LWR@amsmathbody` or `\LWR@amsmathbodynumbered` telling `lateximage` to use as the HTML `<alt>` tag the environment's contents which were automatically captured by the $\mathcal{AMS}$ environment.

For MathJax: Each environment is syched with LaTeX's equation numbers, typeset with LaTeX inside an HTML comment, then printed to HTML output for MathJax to process.

Env `eqnarray` This environmnet is not an $\mathcal{AMS}$ environment and thus its body is not automatically captured, so the **environ** package is used to capture the environment into `\BODY`.

```
\let\LWR@origeqnarray\eqnarray
\let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
\newbool{LWR@numbereqnarray}
\booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from **fancybox**):

```
\newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the LaTeX expression:

```
\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{%
```

If MathJax, the environment contents (the `\BODY`) are executed in a HTML comment to trigger the correct equation number increment (if not starred), then are included verbatim in the output for MathJax to interpret:

```
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \ifbool{LWR@numbereqnarray}
    {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
        \LWR@beginhideamsmath
        \LWR@origeqnarray
        \BODY
        \LWR@origendeqnarray
        \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
        \LWR@addmathjax{eqnarray}{\BODY}
    }%
    {% not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
    \LWR@addmathjax{eqnarray*}{\BODY}
    }% LWR@numbereqnarray
}% mathjax
{% not mathjax
    \ifbool{LWR@numbereqnarray}
    {
```

For numbered SVG equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```
        \begin{BlockClass}{displaymathnumbered}%
        \LWR@newautoidanchor%
        \booltrue{LWR@indisplaymathimage}%
        \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
            \LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```
        \LWR@origeqnarray
        \BODY
        \LWR@origendeqnarray
        \end{lateximage}
        \end{BlockClass}
    }%
    {% not LWR@numbereqnarray
```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```
        \begin{BlockClass}{displaymath}
        \LWR@newautoidanchor%
        \booltrue{LWR@indisplaymathimage}%
        \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
        \LWR@origeqnarray
        \BODY
        \nonumber
        \LWR@origendeqnarray
        \end{lateximage}
        \end{BlockClass}
    }% LWR@numbereqnarray
}% not mathjax
```

Default to number equations in the future:

```
\booltrue{LWR@numbereqnarray}
}
```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```
\RenewEnviron{eqnarray}
{%

\LWR@eqnarrayfactor

}
```

The starred version is patched to turn off the numbering:

```
\csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

```
\end{warpHTML}
```

76 Lateximages

76.1 Description

Env `lateximage` A `lateximage` is a piece of the document which is typeset in LaTeX then included in the HTML output as an image. This is used for math if SVG math is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 74.)

A `lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for **lwarpmk** to extract the `lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML `<span>` is created to hold both the HTML comment, which will have the **pdftotext** conversion, and also the link to the final `.svg` image.

A LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

HTML `alt` tag The HTML `alt` tag is set to the LaTeX source for SVG math, some chemistry expressions, and perhaps some other expressions which make sense for text copy/paste. In some other cases, the `alt` tag is set according to the package name.

When creating an SVG math image, its `alt` tag may be set to the math expression, which may be hashed for image reuse. In the case of `\ensuremath` or after `\StartDynamicMath`, where the contents require a unique image for each instance of the same expression, the `alt` tag is set to `\mathimagename`, and the image is not reused.

This expression is visible in the browser if images are not loaded, and appears when the text is copied and pasted. The default is “math image”, and it may be changed

according to the document's language. This may be set in the preamble, or changed as necessary inside the document, where it will affect the following SVG math images.

For many packages, the output is placed inside a `lateximage` with an HTML `alt` tag set to the package name followed by `\packagediagramname`. For example:

```
(-xy- diagram)
```

This expression is visible in the browser if images are not loaded, and appears when the text is copied and pasted. The default is "diagram", and may it be changed according to the document's language. This may be set in the preamble, or changed as necessary inside the document, where it will affect the following `lateximages`.

SVG image font size

For the `lateximage` environment, the size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, which defaults to:

```
\renewcommand{\LateximageFontSizeName}{normalsize}
```

For inline SVG math, font size is instead controlled by `\LateximageFontScale`, which defaults to:

```
\newcommand*{\LateximageFontScale}{.75}
```

76.2 Support counters and macros

for HTML output:

```
\begin{warpHTML}
```

Ctr `LWR@lateximagenumber`

Sequence the images.

```
\newcounter{LWR@lateximagenumber}
\setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth`

Do not create `\lateximage` inside of `\lateximage`.

```
\newcounter{LWR@lateximagedepth}
\setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
\edef\LWR@hashmark{\string#} % for use in \write
\edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@LIpage`

Used to reference the PDF page number of a lateximage to be written into `lateximages.txt`.

```
\newcounter{LWR@LIpage}
```

```
\end{warpHTML}
```

76.3 Font size

for HTML & PRINT:

```
\begin{warpall}
```

`\LateximageFontSizeName` Declares how large to write text in `\lateximages`. The `.svg` file text size should blend well with the surrounding HTML text size.

⚠ no backslash *Do not include the leading backslash in the name.*

```
\newcommand*{\LateximageFontSizeName}{normalsize}
```

`\LateximageFontScale` Declares how large to scale inline SVG math images. The `.svg` file text size should blend well with the surrounding HTML text size. The default is .75, but it may be redefined as needed depending on the HTML font.

```
\newcommand*{\LateximageFontScale}{.75}
```

```
\end{warpall}
```

76.4 Sanitizing math expressions for HTML

for HTML output:

```
\begin{warpHTML}
```

`\LWR@HTMLsanitize` {⟨*text*⟩}

Math expressions are converted to `lateximages`, and some math environments may contain `&`, `<`, or `>`, which should not be allowed inside an HTML `<alt>` tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way…

```
\newrobustcmd{\LWR@HTMLsanitize}[1]{%
```

Cancel French **babel** character handling, and fully expand the strings:

```
\begingroup%
\LWR@FBcancel%
\fullexpandarg%
```

The &, <, and > may be interpreted by the browser:

```
\protect\StrSubstitute{\detokenize{#1}}%
{\detokenize{&}}{\detokenize{&amp;}}[\LWR@strresult]%

\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{<}}{\detokenize{&lt;}}[\LWR@strresult]%

\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{>}}{\detokenize{&gt;}}[\LWR@strresult]%
```

The double quote occasionally causes problems.

```
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{"}}{\detokenize{&quot;}}[\LWR@strresult]%
```

MathJax allows expressions to be defined with `\newcommand`. These expressions would appear with ## for each argument, and each must be changed to a single #. This must be done after all the above changes. Attempting another conversion after this causes an error upon further expansion.

```
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{##}}{\LWR@origpound}[\LWR@strresult]%

\LWR@strresult%
\endgroup%
}
```

`\LWR@HTMLsanitizeexpand` {⟨*text*⟩}

This version expands the argument before sanitizing it.

```
\newrobustcmd{\LWR@HTMLsanitizeexpand}[1]{%
```

Cancel French **babel** character handling, and fully expand the strings:

```
\begingroup%
\LWR@FBcancel%
\fullexpandarg%
```

The difference between this and `\LWR@HTMLsanitize` (without "expand") is the following `\expandafter`:

```
\protect\StrSubstitute{\detokenize\expandafter{#1}}%
{\detokenize{&}}{\detokenize{&amp;}}[\LWR@strresult]%
```

```
\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{<}}{\detokenize{&lt;}}[\LWR@strresult]%

\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{>}}{\detokenize{&gt;}}[\LWR@strresult]%

\protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
{\detokenize{"}}{\detokenize{&quot;}}[\LWR@strresult]%
```

\LWR@HTMLsanitizeexpand is not used for defining new MathJax macros, so the ## conversion is not needed here.

```
\LWR@strresult%
\endgroup%
}
```

76.5 Equation numbers

Ctr LWR@startingequation

For use with lateximage and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```
\newcounter{LWR@startingequation}

\@ifundefined{chapter}
{
\renewcommand{\theLWR@startingequation}{%
\arabic{LWR@startingequation}%
}
}
{% chapter defined
\renewcommand{\theLWR@startingequation}{%
\ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}%
\arabic{LWR@startingequation}%
}
}
```

Bool LWR@isstartingequation

True for the first equation tag, false for later tags in the same environment.

```
\newbool{LWR@isstartingequation}
```

\LWR@startingequationtag

Prints the starting equation number or tag.

```
\let\LWR@startingequationtag\theLWR@startingequation
```

\LWR@equationtag Prints the ending equation number or tag.

This is reset by lateximage, may be temporarily overwritten by \tag calling \LWR@remembertag.

```
\newcommand*{\LWR@equationtag}{}
```

Only if SVG math, patch \tag after packages have loaded, in case someone else modified \tag.

```
\AtBeginDocument{

\ifbool{mathjax}{}{% not mathjax
```

\LWR@remembertag {⟨*tag*⟩}

For use inside the math environments while using SVG math. Sets \theLWR@startingequation and \theequation to the given tag.

```
\NewDocumentCommand{\LWR@remembertag}{m}{%
\ifbool{LWR@isstartingequation}%
{%
    \global\boolfalse{LWR@isstartingequation}%
    \xdef\LWR@startingequationtag{#1}%
}{}%
\xdef\LWR@equationtag{#1}%
}%

}% not mathjax
}% AtBeginDocument
```

76.6 HTML alt tags

\LWR@amsmathbody {⟨*envname*⟩} For use inside the optional argument to a lateximage to add the contents of a AMS math environment to the <alt> tag.

```
\newcommand*{\LWR@amsmathbody}[1]
{%
\textbackslash\{begin\}\{#1\} % extra space
\LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
\textbackslash\{end\}\{#1\}%
}
```

\LWR@amsmathbodynumbered {⟨*envname*⟩} For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `alt` tag, prefixed by the equation numbers.

```
\newcommand*{\LWR@amsmathbodynumbered}[1]
{%
\ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}%
{(\LWR@equationtag)}%
{(\LWR@startingequationtag--\LWR@equationtag)} % extra space
\LWR@amsmathbody{#1} % extra space
}
```

76.7 `lateximage` environment

Env `lateximage` * [⟨*2: <alt> tag*⟩] [⟨*3: add'l hashing*⟩] [⟨*4: CSS style*⟩]

Typesets the contents and then renders the result as an SVG file. Star causes the image to be hashed for reuse.

The optional `<alt>` tag is included in the HTML code for use with copy/paste.

image filename hashing If starred, a hashed filename is used. If so, the hash is based on the `alt` tag and also the additional hashing argument.

This may be used to provide an expression with a simple `alt` tag but also enough additional information to provide a unique hash.

An example is when the expression is a complicated TEX expression, which would not copy/paste well. A simplified tag may be used, while the complicated expression is duplicated in the additional hashing argument.

Another example is when the expression is simple, but the image depends on options. These options may be decoded into text form and included in the additional hashing argument in order to make the hash unique according to the set of options, even if the simple `alt` tag is still the same.

```
\catcode`\$=\active%

\NewDocumentEnvironment{lateximage}{s O{(image)} O{} O{}}
{%
\LWR@traceinfo{lateximage: starting on \jobname.pdf page \arabic{page}}%
\LWR@traceinfo{lateximage: entering depth is \arabic{LWR@lateximagedepth}}%
```

Nested `lateximages` remain one large `lateximage`:

```
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
```

If nesting inside an already-existing lateximage, simply record one more level. $\mathcal{AMS}$ packages redefine \addtocounter to do nothing if inside a \text, so lower-level TEX macros are used for tracking nested lateximages.

```
{%
%     \addtocounter{LWR@lateximagedepth}{1}%
    \global\advance\c@LWR@lateximagedepth 1\relax% Due to AmS \text macro.
}%
```

Otherwise, this is the outer-most lateximage:

```
{% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
\LWR@traceinfo{lateximage: starting outer-most lateximage}%
    \setcounter{LWR@startingequation}{\value{equation}}%
    \addtocounter{LWR@startingequation}{1}%
    \booltrue{LWR@isstartingequation}%
    \let\LWR@startingequationtag\theLWR@startingequation%
```

The default equation tag, unless overwritten by \tag:

```
    \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
    \addtocounter{LWR@lateximagenumber}{1}%
    \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
    \boolfalse{mathjax}%
```

Be sure that are doing a paragraph:

```
    \LWR@ensuredoingapar%
```

Next file:

```
    \addtocounter{LWR@externalfilecnt}{1}%
    \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be. \setcounterpageref assigns LWR@LIpage to the page number for the reference LWRlateximageXXX:

```
    \setcounterpageref{LWR@LIpage}{LWRlateximage\arabic{LWR@lateximagenumber}}%
    \LWR@traceinfo{lateximage: LWR@LIpage is \arabic{LWR@LIpage}}%
```

Create an HTML span which will hold the comment which contains the **pdftotext** translation of the image's page, and also will hold the link to the `.svg` file:

```
    \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" % extra space
    class="lateximagesource"}%
```

Write instructions to the `lateximages.txt` file:

```
    \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
    \IfBooleanTF{#1}% starred
    {% hash
        \LWR@traceinfo{lateximage: hash true, adding %
            !\detokenize\expandafter{#2}!\detokenize\expandafter{#3}!}%
```

Compute and save the hashed file name for later use:

```
        \edef\LWR@hashedname{%
            \LWR@mdfive{\detokenize\expandafter{#2}-!-#3}%
%             \LWR@mdfive{\detokenize\expandafter{#2}-!-\detokenize\expandafter{#3}}%
        }%
        \LWR@traceinfo{lateximage: hash is \LWR@hashedname}%
```

Write the page, hashing, and hashed name:

```
        \immediate\write\LWR@lateximagesfile{%
            |\arabic{LWR@LIpage}|true|\LWR@hashedname|%
        }%
    }% hash
    {% no hash
```

No hash, so write the page, no hashing, and the image number:

```
        \LWR@traceinfo{lateximage: hash false}%
        \immediate\write\LWR@lateximagesfile{%
            |\arabic{LWR@LIpage}|false|lateximage-\arabic{LWR@externalfilecnt}|%
        }%
    }% no hash
```

Place an open comment tag. This will hide any traces of the lateximage PDF page which were picked up by **pdftotext**.

```
    \LWR@traceinfo{lateximage: about to create open comment}%
    \LWR@htmlopencomment%
```

One level deeper. At this outer-most lateximage, it is known that this is not being used inside an $\mathcal{AMS}$ \text, since the outer-most level will never be in math mode.

```
9313      \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
9314    \LWR@traceinfo{lateximage: about to create a new page}%
9315    \LWR@orignewpage%
```

Typeset the image in a "standard" width page and font size:

```
9316    \LWR@traceinfo{lateximage: about to create minipage}%
9317    \LWR@print@minipage{6in}%
9318    \@nameuse{LWR@print@\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
9319    \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
9320    \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
9321    \def\@mpfn{footnote}%
9322    \def\thempfn{\thefootnote}%
9323    \LetLtxMacro\@footnotetext\LWR@footnotetext%
```

Create the LWRlateximage<number> label:

```
9324    \LWR@traceinfo{lateximage: about to create label}%
9325    \LWR@orig@label{LWRlateximage\arabic{LWR@lateximagenumber}}%
9326    \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
9327    \LetLtxMacro$\LWR@origdollar%
9328    \catcode`\$=3% math shift
9329    \LetLtxMacro\(\LWR@origopenparen%
9330    \LetLtxMacro\)\LWR@origcloseparen%
```

Only enable print-mode display math if are not already inside display math:

```
9331    \ifbool{LWR@indisplaymathimage}{}{% not in display math
9332        \LetLtxMacro\[\LWR@origopenbracket%
9333        \LetLtxMacro\]\LWR@origclosebracket%
```

```
        \let\equation\LWR@origequation%
        \let\endequation\LWR@origendequation%
        \csletcs{equation*}{LWR@origequation*}%
        \csletcs{endequation*}{LWR@origendequation*}%
    }% not in display math
```

For **chemformula**:

```
    \LetLtxMacro\LWR@newsingledollar$%
    \LetLtxMacro\LWR@newsingledollar$% syntax highlighting

}% end of outer-most lateximage
\LWR@traceinfo{lateximage: finished start of environment}%
}% end of \begin{lateximage}
```

\endlateximage When the environment closes:

```
{% start of \end{lateximage}
\LWR@traceinfo{lateximage: starting end of lateximage}%
```

Nested more than one deep?

```
\LWR@traceinfo{lateximage: internal depth was \arabic{LWR@lateximagedepth}}%
\ifnumcomp{\value{LWR@lateximagedepth}}{>}{1}%
```

If nesting inside an already existing lateximage, simply record one less level. Uses a lower-level TEX macro due to $\mathcal{AMS}$ \text change of \addtocounter.

```
{%
    \LWR@traceinfo{lateximage: unnesting}%
    \global\advance\c@LWR@lateximagedepth -1\relax%
}%
```

If this is the outer-most lateximage:

```
{% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
\LWR@traceinfo{lateximage: ending outer-most lateximage}%
    \endLWR@print@minipage%
    \LWR@orignewpage%
    \LWR@print@scriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by **pdftotext**:

```
    \LWR@print@vspace*{.5\baselineskip}%
    \LWR@htmlclosecomment%
    \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

```
    \IfBooleanTF{#1}% starred
    {% hash
        \LWR@subinlineimage[#2]{lateximage}%
        {%
            lateximages\OSPathSymbol%
            \LWR@print@mbox{\LWR@hashedname}%
        }{svg}{#4}%
    }% hash
    {% no hash
        \LWR@subinlineimage[#2]{lateximage}%
        {%
            lateximages\OSPathSymbol%
            \LWR@print@mbox{lateximage-\theLWR@externalfilecnt}%
        }{svg}{#4}%
    }% no hash
```

Be sure that are doing a paragraph:

```
    \LWR@ensuredoingapar%
```

Close the HTML span which has the **pdftotext** comment and also the link to the `.svg` image:

```
    \LWR@htmltag{/span}%
    \ifbool{HTMLDebugComments}{%
        \LWR@htmlcomment{End of lateximage}%
    }{}%
```

Undo one lateximage level. This is not inside an $\mathcal{AMS}$ `\text`, so regular `\addtocounter` may be used here.

```
    \addtocounter{LWR@lateximagedepth}{-1}%
}% end of outer-most lateximage
\LWR@traceinfo{lateximage: exiting depth is \arabic{LWR@lateximagedepth}}%
\LWR@traceinfo{lateximage: done}%
}%
\catcode`\$=3% math shift
\end{warpHTML}
```

for PRINT output:

```
\begin{warpprint}
```

Env `lateximage` [⟨*<alt> tag*⟩] [⟨*CSS style*⟩]

varwidth is used to create a box of the natural width of its contents.

```
\NewDocumentEnvironment{lateximage}{s o o o}
    {\begin{varwidth}[b]{\linewidth}}
    {\end{varwidth}}

\end{warppprint}
```

77 center, flushleft, flushright

for HTML output:

```
\begin{warpHTML}
```

Env center Replace center functionality with CSS tags:

```
\renewenvironment*{center}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[\LWR@print@mbox{text-align:center}]{center}}
{\BlockClass{center}}
}
{\endBlockClass}
```

Env flushright

```
\renewenvironment*{flushright}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[\LWR@print@mbox{text-align:right}]{flushright}}
{\BlockClass{flushright}}
}
{\endBlockClass}
```

Env flushleft

```
\renewenvironment*{flushleft}
{
\LWR@forcenewpage
\ifbool{FormatWP}
{\BlockClass[\LWR@print@mbox{text-align:left}]{flushleft}}
{\BlockClass{flushleft}}
}
{\endBlockClass}
```

\centering, \raggedleft, and \raggedright usually have no effect on the HTML output, but they may be used to compare with the next token to identify their use at the start of a float. See \LWR@floatalignment.

\centering

```
\newcommand*{\LWR@HTML@centering}{%
\ifbool{HTMLDebugComments}{%
    \LWR@htmlcomment{centering}%
}{}%
}
\LWR@formatted{centering}
```

\raggedleft

```
\newcommand*{\LWR@HTML@raggedleft}{%
\ifbool{HTMLDebugComments}{%
    \LWR@htmlcomment{raggedleft}%
}{}%
}
\LWR@formatted{raggedleft}
```

\raggedright

```
\newcommand*{\LWR@HTML@raggedright}{%
\ifbool{HTMLDebugComments}{%
    \LWR@htmlcomment{raggedright}%
}{}%
}
\LWR@formatted{raggedright}
```

\leftline {⟨*text*⟩}

```
\renewcommand{\leftline}[1]{\begin{flushleft}#1\end{flushleft}}
```

\centerline {⟨*text*⟩}

```
\renewcommand{\centerline}[1]{\begin{center}#1\end{center}}
```

\rightline {⟨*text*⟩}

```
\renewcommand{\rightline}[1]{\begin{flushright}#1\end{flushright}}
```

```
\end{warpHTML}
```

78 Pre-loaded packages

for HTML output:
```
\begin{warpHTML}
```

If **textcomp** was loaded before **lwarp**, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```
\@ifpackageloaded{textcomp}{\LWR@origRequirePackage{lwarp-textcomp}}{}
```

If **xunicode** was loaded before **lwarp**, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```
\@ifpackageloaded{xunicode}{\LWR@origRequirePackage{lwarp-xunicode}}{}
```

If **graphics** or **graphicx** were loaded before **lwarp**, perhaps by **xunicode**, explicitly load the lwarp patches now:

```
\@ifpackageloaded{graphics}{\LWR@origRequirePackage{lwarp-graphics}}{}
\@ifpackageloaded{graphicx}{\LWR@origRequirePackage{lwarp-graphicx}}{}
```

```
\end{warpHTML}
```

79 Siunitx

Pkg `siunitx` The **lwarp** core passes a few options to **siunitx**.

fractions Due to **pdftolatex** limitations, fraction output is replaced by symbol output for `per-mode` and `quotient-mode`.

⚠ math mode required Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the **siunitx** extension for MathJax is not currently hosted at any public CDN, thus **siunitx** is not usable with MathJax unless a local copy of this extension is created first.

⚠ `tabular` Tabular `S` columns are rendered as simple `c` columns, and tabular `s` columns are not supported. These may be replaced by `c` columns with each cell contained in `\num` or `\si`.

for HTML output:
```
\begin{warpHTML}
```

Options for siunitx:

```
\newrobustcmd{\LWR@siunitx@textcelsius}{\HTMLentity{deg}C}
```

```
\newrobustcmd{\LWR@siunitx@textdegree}{\HTMLentity{deg}}
\newrobustcmd{\LWR@siunitx@textprime}{\HTMLunicode{2032}}
\newrobustcmd{\LWR@siunitx@textdblprime}{\HTMLunicode{2033}}
\newrobustcmd{\LWR@siunitx@textplanckbar}{\text{\textit{\HTMLunicode{0127}}}}

\appto\LWR@restoreorigformatting{%
\renewrobustcmd{\LWR@siunitx@textcelsius}{\text{\ensuremath{^\circ}C}}%
\renewrobustcmd{\LWR@siunitx@textdegree}{\text{\ensuremath{^\circ}}}%
\renewrobustcmd{\LWR@siunitx@textprime}{\text{\ensuremath{^\prime}}}%
\renewrobustcmd{\LWR@siunitx@textdblprime}{\text{\ensuremath{^{\prime\prime}}}}%
\renewrobustcmd{\LWR@siunitx@textplanckbar}{\text{\ensuremath{\hbar}}}%
}

\PassOptionsToPackage{
    detect-mode=true,
    per-mode=symbol,% fraction is not seen by pdftotext
    text-celsius = {\LWR@siunitx@textcelsius},
    text-degree = {\LWR@siunitx@textdegree},
    text-arcminute = {\LWR@siunitx@textprime} ,
    text-arcsecond = {\LWR@siunitx@textdblprime} ,
}{siunitx}

\end{warpHTML}
```

80 Graphics print-mode modifications

80.1 General limitations

\includegraphics file formats

For `\includegraphics` with `.pdf` or `.eps` files, the user must provide a `.pdf` or `.eps` image file for use in print mode, and also a `.svg`, `.png`, or `.jpg` version of the same image for use in HTML.

```
\includegraphics{filename} % print:.pdf/.eps HTML:.svg, etc.
```

For print output, **lwarp** will automatically choose the `.pdf` or `.eps` format if available, or some other format otherwise. For HTML, one of the other formats is used instead.

If a `.pdf` or `.eps` image is referred to with its file extension, the extension will be changed to `.svg` for HTML:

```
\includegraphics{filename.pdf} % uses .svg in HTML
\includegraphics{filename.eps} % uses .svg in HTML
```

Prog pdftocairo PDF to SVG

To convert a PDF image to SVG, use the utility **pdftocairo**:

Enter ⇒ `pdftocairo -svg filename.pdf`

Prog `lwarpmk pdftosvg` For a large number of images, use **lwarpmk**:

Enter ⇒ `lwarpmk pdftosvg *.pdf` (or a list of filenames)

Prog `lwarpmk epstopdf` For EPS images converted to PDF using the package **epstopdf**, use

Prog `epstopdf` **epstopdf** package

Enter ⇒ `lwarpmk pdftosvg *.PDF`

to convert to SVG images.

DVI latex When using DVI **latex**, it is necessary to convert EPS to PDF and then to SVG:

Enter ⇒ `lwarpmk epstopdf *.eps` (or a list of filenames)

Enter ⇒ `lwarpmk pdftosvg *.pdf` (or a list of filenames)

PNG and JPG For PNG or JPG while using **pdflatex**, **lualatex**, or **xelatex**, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

GIF GIF files may be used for HTML, but another format must also be provided for print output.

file extension priorities If a file extension is not used, for HTML the file extension priorities are: SVG, GIF, PNG, then JPG.

⚠ **graphics** vs. **graphicx** If using the older **graphics** syntax, use both optional arguments for `\includegraphics`. A single optional parameter is interpreted as the newer **graphicx** syntax. Note that

⚠ viewports viewports are not supported by **lwarp** — the entire image will be shown.

units For `\includegraphics`, avoid `px` and % units for width and height, or enclose them inside `warpHTML` environments. For font-proportional image sizes, use `ex` or `em`. For fixed-sized images, use `cm`, `mm`, `in`, `pt`, or `pc`. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area. Do not use the `scale` option, since it is not well supported by HTML browsers.

options `\includegraphics` accepts `width` and `height`, `origin`, `rotate` and `scale`, plus a new `class` key.

HTML class With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

`\rotatebox` `\rotatebox` accepts the optional `origin` key.

⚠ browser support `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support.

The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike LaTeX, so expect some ugly results for scaling and rotating.

80.2 Print-mode modifications

for PRINT output: For print output, accept and then discard the new `class` key:

```
\begin{warpprint}
\define@key{Gin}{class}{}
```

Print-mode additions for the **overpic** package. See section 300 for the HTML version.

```
\AtBeginDocument{
\@ifpackageloaded{overpic}{
\newcommand*{\overpicfontsize}{12}
\newcommand*{\overpicfontskip}{14}
}{}
}
\end{warpprint}
```

81 Xcolor boxes

Pkg `xcolor` A few new definitions are provided for enhanced HTML colored boxes, and `\fcolorbox` is slightly modified. Print-mode version are also provided.

Print-mode versions of new **xcolor** defintions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the **xcolor** originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of `none`, in which case only the frame is drawn, allowing the background page color to show.

for HTML & PRINT:

```
\begin{warpall}
```

After **xparse** may have been loaded …

```
\AtBeginDocument{
```

… and *only* if **xcolor** was loaded:

```
\@ifpackageloaded{xcolor}{
\LWR@traceinfo{patching xcolor}
```

The print version:

\colorboxBlock `\colorboxBlock` is the same as `\colorbox`:

```
\LetLtxMacro\colorboxBlock\colorbox
```

The original definition is reused by the new versions:

```
\LetLtxMacro\LWR@orig@print@fcolorbox\fcolorbox
```

\fcolorbox `[`⟨*framemodel*⟩`] {`⟨*framecolor*⟩`} [`⟨*boxmodel*⟩`] {`⟨*boxcolor*⟩`} {`⟨*text*⟩`}`

In print mode, `\fcolorbox` is modified to accept a background color of `none`.

(`\fcolorbox` is particular about its optional arguments, thus the elaborate combinations of `\ifthenelse`.)

```
\newsavebox{\LWR@colorminipagebox}

\NewDocumentCommand{\LWR@print@fcolorbox}{o m o m +m}{%
\LWR@traceinfo{LWR@print@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
\begin{lrbox}{\LWR@colorminipagebox}%
#5%
\end{lrbox}%
```

Sort out the various optional arguments and the background color of `none`. In each case, the LRbox is placed inside a `\fcolorbox`.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```
\ifthenelse{\equal{#4}{none}}%
{% #4 none
    \LWR@traceinfo{background is none}%
    {% scope the \colorlet
        \colorlet{LWR@currentcolor}{.}%
        \color{#2}%
        \fbox{%
            \color{LWR@currentcolor}%
            \usebox{\LWR@colorminipagebox}%
        }% fbox
    }% colorlet
}% #4 none
{% #4 not none
```

```
\LWR@traceinfo{background not none}%
\IfValueTF{#1}%
{%
    \IfValueTF{#3}%
    {\LWR@orig@print@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWR@orig@print@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
}%
{% no value #1
    \IfValueTF{#3}%
    {\LWR@orig@print@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWR@orig@print@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
}% no value #1
}% #4 not none
\LWR@traceinfo{LWR@print@fcolorbox done}%
}

\renewcommand*{\fcolorbox}{\LWR@print@fcolorbox}
```

`\fcolorboxBlock` `[`⟨*frame model*⟩`]` `{`⟨*frame color*⟩`}` `[`⟨*box model*⟩`]` `{`⟨*box color*⟩`}` `{`⟨*text*⟩`}`

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
\newcommand*{\LWR@print@fcolorboxBlock}{\LWR@print@fcolorbox}
\newcommand*{\fcolorboxBlock}{\LWR@print@fcolorboxBlock}
```

Env `fcolorminipage` `[`⟨*1:frame model*⟩`]` `{`⟨*2:frame color*⟩`}` `[`⟨*3:box model*⟩`]` `{`⟨*4:box color*⟩`}` `[`⟨*5:align*⟩`]` `[`⟨*6:height*⟩`]` `[`⟨*7:inner-align*⟩`]` `{`⟨*8:width*⟩`}`

In print mode, becomes a `\fcolorbox` containing a `minipage`:

```
\NewDocumentEnvironment{LWR@print@fcolorminipage}{o m o m O{c} O{} o m}
{%
\LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
\begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
\IfValueTF{#7}%
{\begin{minipage}[#5][#6][#7]{#8}}%
{\begin{minipage}[#5][#6][#5]{#8}}%
}%
{%
\end{minipage}%
\end{lrbox}%
\LWR@traceinfo{*** starting end fcolorminipage #1 #2 #3 #4 #8}%
```

Sort out the various optional arguments and the background color of `none`. In each case, the LRbox is placed inside a `\fcolorbox`.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```
\ifthenelse{\equal{#4}{none}}%
{% #4 none
    {% scope the \colorlet
        \colorlet{LWR@currentcolor}{.}%
        \color{#2}%
        \fbox{%
            \color{LWR@currentcolor}%
            \usebox{\LWR@colorminipagebox}%
        }% fbox
    }% colorlet
}% #4 none
{% #4 not none
    \IfValueTF{#1}%
    {%
    \IfValueTF{#3}%
    {\LWR@orig@print@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWR@orig@print@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
    }%
    {% no value #1
    \IfValueTF{#3}%
    {\LWR@orig@print@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
    {\LWR@orig@print@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
    }% no value #1
}% #4 not none
\LWR@traceinfo{*** finished end fcolorminipage}%
}

\newenvironment*{fcolorminipage}
    {\LWR@print@fcolorminipage}
    {\endLWR@print@fcolorminipage}


\LWR@traceinfo{xcolor patches done}
}{}% xcolor loaded
}% AtBeginDocument


\end{warpall}
```

82 Chemmacros environments

\makepolymerdelims and redox reactions must be enclosed in a lateximage during HTML output. These environments are provided here in print mode, and in the **chemmacros** code in HTML mode, as a high-level semantic syntax which automatically embeds the contents in a lateximage with an appropriate alt tag.

for PRINT output:

```
\begin{warpprint}

\AtBeginDocument{
\@ifpackageloaded{chemmacros}{
```

Env polymerdelims

```
\DeclareDocumentEnvironment{polymerdelims}{}
    {}{}
```

Env redoxreaction {⟨*space above*⟩} {⟨*space below*⟩}

For print output, extra space is include above and below the image, and a lateximage is not necessary. This extra space must be enforced, even inside a float, so zero-width rules are used.

For the HTML version, see section 155.4.

```
\DeclareDocumentEnvironment{redoxreaction}{m m}
    {\rule{0pt}{#1}}{\rule[-#2]{0pt}{#2}}

}{}% chemmacros
}% AtBeginDocument

\end{warpprint}
```

83 Cleveref

Pkg cleveref **cleveref** package is used as-is with minor patches.

⚠ **cleveref** page numbers **cleveref** and **varioref** are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for \cpageref and \cpagerefrange. This phrase includes \cpagerefFor, which defaults to “for”.

Ex:

```
    \cpageref{tab:first,tab:second}
 in HTML becomes:
```
"pages **for** table 4.1 and **for** table 4.2"

See `\cpagerefFor` at page 539 to redefine the message which is printed for page number references.

loading order

cleveref and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires **cleveref**. The user's document may not require **cleveref**, thus the user may never explicitly load it, so during HTML output **lwarp** loads it last. If the user's document preamble uses **cleveref** options, or functions such as `\crefname`, then **cleveref** may be loaded in the user's preamble near the end, and **lwarp**'s additional loading of **cleveref** will have no effect.

Table 10 on page 455 shows the data structure of the label/reference system as revised by **lwarp** and **cleveref**.

A few patches allow **cleveref** to work as-is:

for HTML output:

```
\begin{warpHTML}
```

`\AtEndPreable` forces **cleveref** to be loaded last:

```
\AtEndPreamble{
\RequirePackage{cleveref}
}
```

The following patches are applied after **cleveref** has loaded, and after `\AtBeginDocument`. Print-mode versions are not required since they all come down to `\ref` eventually, and `\ref` has a print-mode version.

```
\AfterEndPreamble{
\LWR@traceinfo{Patching cleveref.}
```

`\@@@setcref` {⟨*kindofref*⟩} {⟨*label*⟩}

`\@templabel` becomes the section number.

```
\def\LWR@orig@@@setcref#1#2{\cref@getlabel{#2}{\@templabel}#1{\@templabel}{}{}}%

\ifdefequal{\@@setcref}{\LWR@orig@@@setcref}{% before v0.21
    \renewcommand*{\@@setcref}[2]{#1{\ref{#2}}{}{}}
}{
    \ifdefequal{\@@@setcref}{\LWR@orig@@@setcref}{% as of v0.21
```

```
        \renewcommand*{\@@@setcref}[2]{#1{\ref{#2}}{}{}}
    }{
        \PackageWarning{lwarp-cleveref}{
            Unknown version of cleveref.
            \protect\cref\space will fail.
        }%
    }
}
```

`\@@@setcrefrange` {⟨*text*⟩} {⟨*label*⟩} {⟨*label*⟩}

```
\def\LWR@orig@@@setcrefrange#1#2#3{%
  \cref@getlabel{#2}{\@labela}%
  \cref@getlabel{#3}{\@labelb}%
  #1{\@labela}{\@labelb}{}{}{}{}}%

\ifdefequal{\@@setcrefrange}{\LWR@orig@@@setcrefrange}{
    \renewcommand{\@@setcrefrange}[3]{%
        #1{\ref{#2}}{\ref{#3}}{}{}{}{}%
    }
}{
    \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{
        \renewcommand{\@@@setcrefrange}[3]{%
            #1{\ref{#2}}{\ref{#3}}{}{}{}{}%
        }
    }{
        \PackageWarning{lwarp-cleveref}{
            Unknown version of cleveref.
            \protect\crefrange\space will fail.
        }
    }
}

```

`\cpagerefFor` Redefinable word between "page(s)" and the page numbers.

```
\newcommand*{\cpagerefFor}{for}
```

`\@@@setcpageref` {⟨*typeofref*⟩} {⟨*label*⟩}, where typeofref is "page" or "pages"

```
\def\LWR@orig@@setcpageref#1#2{% before v0.21
  \cref@getpageref{#2}{\@temppage}#1{\@temppage}{}{}}%

\def\LWR@orig@@@setcpageref#1#2{% as of v0.21
  \cpageref@getlabel{#2}{\@temppage}#1{\@temppage}{}{}}%

```

```
\ifdefequal{\@@setcpageref}{\LWR@orig@@setcpageref}{
    \renewcommand*{\@@setcpageref}[2]{%
        #1{\cpagerefFor\ \cref{#2}}{}{}%
    }
}{
    \ifdefequal{\@@@setcpageref}{\LWR@orig@@@setcpageref}{
        \renewcommand*{\@@@setcpageref}[2]{%
            #1{\cpagerefFor\ \cref{#2}}{}{}%
        }
    }
    {
        \PackageWarning{lwarp-cleveref}{
            Unknown version of cleveref.
            \protect\cpageref\space will fail.
        }
    }
}

\def\LWR@orig@@setcpagerefrange#1#2#3{% before v0.21
  \cref@getpageref{#2}{\@pagea}%
  \cref@getpageref{#3}{\@pageb}%
  #1{\@pagea}{\@pageb}{}{}{}{}}%

\def\LWR@orig@@@setcpagerefrange#1#2#3{% as of v0.21
  \cpageref@getlabel{#2}{\@pagea}%
  \cpageref@getlabel{#3}{\@pageb}%
  #1{\@pagea}{\@pageb}{}{}{}{}}%

\ifdefequal{\@@setcpagerefrange}{\LWR@orig@@setcpagerefrange}{
    \renewcommand*{\@@setcpagerefrange}[3]{%
        #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}%
    }
}{
    \ifdefequal{\@@@setcpagerefrange}{\LWR@orig@@@setcpagerefrange}{
        \renewcommand*{\@@@setcpagerefrange}[3]{%
            #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}%
        }
    }
    {
        \PackageWarning{lwarp-cleveref}{
            Unknown version of cleveref.
            \protect\cpagerefrange\space will fail.
        }
    }
}

}% AfterEndPreamble
```

Remember and patch some label-related defintions. These will be further encased and patched by other packages later.

\label and \pageref do NOT change their behavior according to print or HTML output, and thus do not use the \LWR@formatted system.

```
\LetLtxMacro\LWR@orig@label\label
\RenewDocumentCommand{\label}{}{\LWR@new@label}

\LetLtxMacro\LWR@orig@pageref\pageref
\RenewDocumentCommand{\pageref}{}{\LWR@new@pageref}
\end{warpHTML}
```

84 Picture

Env picture The picture environment is enclosed inside a \lateximage.

for HTML output:

```
\begin{warpHTML}
```

Env picture

```
\BeforeBeginEnvironment{picture}{\begin{lateximage}[(picture)]}

\AfterEndEnvironment{picture}{\end{lateximage}}

\end{warpHTML}
```

85 Boxes and Minipages

A css flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

⚠ inline A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made inline-block.

placement Minipages and parboxes will be placed side-by-side in HTML unless you place a \newline between them.

side-by-side Side-by-side minipages may be separated by \quad, \qquad, \enskip, \hspace, \hfill, or a \rule. When inside a center environment, the result is similar in print

and HTML. Paragraph tags are surpressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

in a span — There is limited support for minipages inside an HTML `<span>`. An HTML `<div>` cannot appear inside a `<span>`. While in a `<span>`, minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an "inline" format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.

size — When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.

no-width minipages — A minipage of width exactly `\linewidth` is automatically given no HTML width.

full-width minipages — A new macro `\minipagefullwidth` requests that the next minipage be generated without an HTML `width` attribute, allowing it to be the full width of the display rather than the fixed width given.

⚠ text alignment — Nested minipages adopt their parent's text alignment in HTML, whereas in regular LaTeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

for HTML output:
```
9685 \begin{warpHTML}
```

85.1 Counters and lengths

Ctr `LWR@minipagedepth` — Used to only reset the line width at the outermost minipage.

```
9686 \newcounter{LWR@minipagedepth}
9687 \setcounter{LWR@minipagedepth}{0}
```

Len `\LWR@minipagewidth` — Used to convert the width into printable units.

```
9688 \newlength{\LWR@minipagewidth}
```

Len `\LWR@minipageheight` — Used to convert the height into printable units.

```
9689 \newlength{\LWR@minipageheight}
```

85.2 Footnote handling

Also see section 55 for other forms of footnotes. Minipage footnotes are gathered in section 55.5, and then placed into the document in section 85.3.

85.3 Minipage handling

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML `width`?

```
\newbool{LWR@minipagefullwidth}
\boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no `width` tag in HTML:

for HTML output:
```
\newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
\end{warpHTML}
```

for PRINT output:
```
\begin{warpprint}
\newcommand*{\minipagefullwidth}{}
\end{warpprint}
```

for HTML output:
```
\begin{warpHTML}
```

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
\newbool{LWR@minipagethispar}
\boolfalse{LWR@minipagethispar}
```

Env `minipage` `[`⟨*vert position*⟩`]` `[`⟨*height*⟩`]` `[`⟨*inner vert position*⟩`]` `{`⟨*width*⟩`}`

The vertical positions may be 'c', 't', or 'b'. The inner position may also be 's'.

When using `\linewidth`, `\textwidth`, or `\textheight`, these are scaled proportionally to a 6×9 inch text area.

```
\NewDocumentEnvironment{LWR@HTML@minipage}{O{t} o O{t} m}
{%
\LWR@traceinfo{minipage}%
```

Temporarily open a group, in which width and height is computed based on a virtual page size instead of the extra-large PDF page used during HTML tag generation.

The following used to be an actual LaTeX minipage.

```
\begingroup
```

Compute width, adjusted for frames:

```
\setlength{\LWR@minipagewidth}{#4}%
\ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
```

Only create a new page if not yet nested:

```
    \LWR@orignewpage%
```

Adjust virtual page size:

```
    \addtolength{\LWR@minipagewidth}{3em}% room for frames
    \setlength{\linewidth}{6in}%
    \setlength{\textwidth}{6in}%
    \setlength{\textheight}{9in}%
}{}%
\LWR@traceinfo{computed width is \LWR@printlength{\LWR@minipagewidth}}%
```

Compute height:

```
\setlength{\LWR@minipageheight}{\textheight}% default unless specified
\IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

Track nesting depth:

```
\addtocounter{LWR@minipagedepth}{1}%
```

LaTeX wants to start a paragraph for the virtual minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
\ifbool{FormatWP}{\newline}{}%
\LWR@stoppars%
```

If `FormatWP`, add a text frame:

```
\ifbool{FormatWP}{%

\addtocounter{LWR@thisautoidWP}{1}%
\LWR@htmltag{%
    div id="\LWR@print@mbox{autoidWP-\arabic{LWR@thisautoidWP}}" %
    class="wpminipage"%
}%

}{}%
```

Create the `<div>` tag with optional alignment style:

```
\LWR@traceinfo{minipage: creating div class}%
\LWR@htmltag{div class="minipage" style="%
\ifthenelse{\equal{#1}{t}}{\LWR@print@mbox{vertical-align:bottom} ; }{}%
\ifthenelse{\equal{#1}{c}}{\LWR@print@mbox{vertical-align:middle} ; }{}%
\ifthenelse{\equal{#1}{b}}{\LWR@print@mbox{vertical-align:top} ; }{}%
```

```
\ifthenelse{\equal{#3}{t}}{\LWR@print@mbox{justify-content:flex-start} ; }{}%
\ifthenelse{\equal{#3}{c}}{\LWR@print@mbox{justify-content:center} ; }{}%
\ifthenelse{\equal{#3}{b}}{\LWR@print@mbox{justify-content:flex-end} ; }{}%
\ifthenelse{\equal{#3}{s}}{\LWR@print@mbox{justify-content:space-between} ; }{}%
```

Print the width and optional height styles:

```
\LWR@traceinfo{minipage: about to print the width of \LWR@printlength{\LWR@minipagewidth}}%
\ifbool{LWR@minipagefullwidth}%
{\boolfalse{LWR@minipagefullwidth}}%
{%
    \ifthenelse{\lengthtest{#4}=\linewidth}%
    {}%
    {width:\LWR@printlength{\LWR@minipagewidth} ; }%
}%
\LWR@traceinfo{minipage: about to print the height}%
\IfValueT{#2}{height:\LWR@printlength{\LWR@minipageheight} ; }%
"}%
```

Finish with an empty line to start the contents on a new line.

```

% The preceding empty line is required.
```

Set the user-accessible line and text width and height values inside the virtual minipage. These do not affect the actual size of the PDF output, but are used by any reference to `\linewidth`, etc. inside the virtual minipage being created here.

```
\setlength{\linewidth}{#4}% the original width
\setlength{\textwidth}{6in}%
\setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
\LWR@print@raggedright%
```

Set minipage footnotes:

```
\def\@mpfn{mpfootnote}%
\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
\let\@footnotetext\@mpfootnotetext%
```

Resume paragraph tag handling for the contents of the minipage:

```
\LWR@startpars%
\ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%

=== begin minipage ===
```

```
}{}%
\LWR@traceinfo{minipage: finished starting the minipage}%
}% finished \minipage
{% \endminipage
```

Print pending minipage footnotes:

```
\LWR@printpendingmpfootnotes%
```

End the environment with closing tag:

```
\ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%

=== end minipage ===

}{}%
\LWR@stoppars%
```

The following used to be an actual LaTeX minipage.

```
\endgroup%

\ifbool{FormatWP}{%

\LWR@htmlelementend{div}%

}{}%
\LWR@htmldivclassend{minipage}%

\addtocounter{LWR@minipagedepth}{-1}%
\LWR@startpars%
\ifbool{FormatWP}{\newline}{}%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
\global\booltrue{LWR@minipagethispar}%
\LWR@traceinfo{LWR@minipage: done}%
}

\LWR@formattedenv{minipage}
```

85.4 Parbox, mbox, makebox, framebox, fbox, raisebox

for HTML output:

\parbox [⟨*pos*⟩] [⟨*height*⟩] [⟨*inner-pos*⟩] {⟨*width*⟩} {⟨*text*⟩}

A parbox uses the minipage code:

```
\NewDocumentCommand{\LWR@HTML@parbox}{O{t} o O{t} m +m}
{
\LWR@traceinfo{parbox of width #4}%
\begin{minipage}[#1][#2][#3]{#4}%
#5
\end{minipage}%
}

\LWR@formatted{parbox}
```

\mbox {⟨*text*⟩} Nullified for HTML.

```
\newcommand*{\LWR@HTML@mbox}[1]{#1}

\LWR@formatted{mbox}
```

\makebox (⟨*(*⟩)posn) [⟨*width*⟩] [⟨*pos*⟩] {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@HTML@makebox}{d() o o m}{%
```

Check for the optional width:

```
\IfValueTF{#2}%
{%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is `justified` alignment.

```
    {% scope
    \def\LWR@align{center}%
    \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
    \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
    \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the `width` argument:

```
    \setlength{\LWR@tempwidth}{#2}%
```

`inline-block` allows `width` and `text-alignment` to be used in a `<span>`.

```
    \InlineClass[%
        \LWR@print@mbox{display:inline-block} ; %
```

```
        \LWR@print@mbox{text-align}:\LWR@align\ ; %
        width:\LWR@printlength{\LWR@tempwidth}%
    ]%
    {makebox}%

    {#4}%
    }% scope
}%
```

Without a width argument, the text is simply used inline:

```
{#4}% no width
}

\LWR@formatted{makebox}
```

`\framebox` `[⟨width⟩] [⟨pos⟩] {⟨text⟩}`

```
\LetLtxMacro\LWR@origframebox\framebox

\NewDocumentCommand{\LWR@HTML@framebox}{o o m}{%
\fbox{\makebox[#1][#2]{#3}}%
}

\LWR@formatted{framebox}
```

`\LWR@forceminwidth` `{⟨legth⟩}`

Sets `\LWR@atleastonept` to be at least 1pt.

```
\newlength{\LWR@atleastonept}

\newcommand*{\LWR@forceminwidth}[1]{%
\setlength{\LWR@atleastonept}{#1}%
\ifthenelse{%
    \lengthtest{\LWR@atleastonept>0pt}\AND%
    \lengthtest{\LWR@atleastonept<1pt}%
}%
{\setlength{\LWR@atleastonept}{1pt}}%
{}%
}
```

`\LWR@blackborderpadding` Prints the HTML attributes for a black border and padding.

`\LWR@forceminwidth` must be used first in order to set the border width.

```
\newcommand*{\LWR@blackborderpadding}{%
border:\LWR@printlength{\LWR@atleastonept} solid black ; %
padding:\LWR@printlength{\fboxsep}%
}
```

`\fbox` `{`⟨*text*⟩`}`

Creates a framed inline span enclosing the text.

Create a new HTML version, but don't use it until after **xcolor** may have loaded:

```
\newcommand{\LWR@HTML@fbox}[1]{%
\LWR@traceinfo{HTML fbox}%
\LWR@forceminwidth{\fboxrule}%
\InlineClass[%
\LWR@blackborderpadding%
]{fbox}{#1}
}
```

xcolor `\let`s things to `\fbox` when it is loaded, and this must remain even for HTML output while in a `lateximage`, so `\fbox` is not modified until `\AtBeginDocument`:

```
\AtBeginDocument{\LWR@formatted{fbox}}
```

`\fboxBlock` `{`⟨*text*⟩`}` Creates a framed HTML `<div>` of the text.

First, a print-mode version. This is newly defined for print mode, so it is defined inside `warpall`.

for HTML & PRINT:

```
\end{warpHTML}
\begin{warpall}
\let\fboxBlock\fbox
\end{warpall}
\begin{warpHTML}
```

for HTML output: Next, an HTML version:

```
\newcommand{\LWR@HTML@fboxBlock}[1]{%
\LWR@forceminwidth{\fboxrule}%
\begin{BlockClass}[%
\LWR@blackborderpadding%
]{fboxBlock}
#1
\end{BlockClass}
}

\LWR@formatted{fboxBlock}

\end{warpHTML}
```

Env `fminipage` [⟨*align*⟩] [⟨*height*⟩] [⟨*align*⟩] {⟨*width*⟩}

Creates a framed HTML around its contents.

for HTML & PRINT: Print version:

```
\begin{warpall}

\newsavebox{\LWR@fminipagebox}

\NewDocumentEnvironment{LWR@print@fminipage}{O{t} o O{t} m}
{%
```

An outer minipage will be used for vertical alignment. An inner minipage will be framed with \fbox.

If the optional inner alignment is not given, use the outer instead:

```
\IfValueTF{#3}%
{\def\LWR@thisalign{#3}}
{\def\LWR@thisalign{#1}}%
```

Form the outer minipage depending on whether a height was given. Make the outer minipage larger to compensate for the frame.

```
\IfValueTF{#2}%
{\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
{\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
\begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
\IfValueTF{#2}%
{\minipage[#1][#2][\LWR@thisalign]{#4}}%
{\minipage[#1]{#4}}%
}
{%
```

Close the inner minipage and the LR box with the contents:

```
\endminipage%
\end{lrbox}%
```

Create a frame around the contents of the environment:

```
\fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
\endminipage%
}

\newenvironment{fminipage}{\LWR@print@fminipage}{\endLWR@print@fminipage}

\end{warpall}
```

for HTML output:

```
\begin{warpHTML}

\NewDocumentEnvironment{LWR@HTML@fminipage}{O{t} o O{t} m}
{%
\LWR@traceinfo{fminipage #1 #2 #3 #4}%
\LWR@forceminwidth{\fboxrule}%
\setlength{\LWR@tempwidth}{#4}%
\IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
\begin{BlockClass}[%
\LWR@blackborderpadding ; %
\IfValueT{#2}{height:\LWR@printlength{\LWR@tempheight} ; }%
width:\LWR@printlength{\LWR@tempwidth}%
]{fminipage}%
}
{%
\end{BlockClass}%
\LWR@traceinfo{fminipage done}%
}

\LWR@formattedenv{fminipage}

\end{warpHTML}
```

`\raisebox` `{⟨raiselen⟩}` `[⟨height⟩]` `[⟨depth⟩]` `{⟨text⟩}`

```
\begin{warpHTML}

\NewDocumentCommand{\LWR@HTML@raisebox}{m o o m}{%
#4%
}

\LWR@formatted{raisebox}

\end{warpHTML}
```

86 Direct formatting

⚠ \bfseries, etc. \textbf, etc. are supported, but \bfseries, etc. work only in some situations.

⚠ HTML special chars &, <, and > have special meanings in HTML. If \&, \textless, and \textgreater are used, the proper result should occur in HTML, but there may be HTML parsing problems if these special characters occur unescaped in program listings or other verbatim text.

For high-level block and inline custom CSS classes, see section 49.8.

for HTML output:

```
\begin{warpHTML}
```

\LWR@HTMLtextstyle {⟨*FormatWP style*⟩} {⟨*class*⟩} {⟨*text*⟩}

If FormatWP, adds an explicit style to the text span class. This is used by LibreOffice to mark its imported text using the given style.

```
\DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
\ifbool{FormatWP}%
{\LWR@htmlspanclass[#1]{#2}{#3}}%
{\LWR@htmlspanclass{#2}{#3}}%
}
```

\emph {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
\DeclareRobustCommand{\LWR@nullemph}[1]{#1}
\LetLtxMacro\emph\LWR@HTMLemph
```

\textmd {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
\LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}%
}
\DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}

\LetLtxMacro\textmd\LWR@HTMLtextmd
```

\textbf {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
\DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
\LetLtxMacro\textbf\LWR@HTMLtextbf
```

\textrm {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
\LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}

\LetLtxMacro\textrm\LWR@HTMLtextrm
```

\textsf {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
\LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
}
\DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
\LetLtxMacro\textsf\LWR@HTMLtextsf
```

\texttt {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
\DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
\LetLtxMacro\texttt\LWR@HTMLtexttt
```

\textup {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
\LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextup}[1]{#1}

\LetLtxMacro\textup\LWR@HTMLtextup
```

\textit {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
\DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
\LetLtxMacro\textit\LWR@HTMLtextit
```

\textsc {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
\LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
}

```

```
\DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}

\LetLtxMacro\textsc\LWR@HTMLtextsc
```

`\textsl` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
\LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
}

\DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}

\LetLtxMacro\textsl\LWR@HTMLtextsl
```

`\textnormal` {⟨*text*⟩}

```
\DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
\DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
\LetLtxMacro\textnormal\LWR@HTMLtextnormal

\DeclareRobustCommand{\LWR@nullrmfamily}{}
\DeclareRobustCommand{\LWR@nullsffamily}{}
\DeclareRobustCommand{\LWR@nullttfamily}{}
\DeclareRobustCommand{\LWR@nullbfseries}{}
\DeclareRobustCommand{\LWR@nullmdseries}{}
\DeclareRobustCommand{\LWR@nullupshape}{}
\DeclareRobustCommand{\LWR@nullslshape}{}
\DeclareRobustCommand{\LWR@nullscshape}{}
\DeclareRobustCommand{\LWR@nullitshape}{}
\DeclareRobustCommand{\LWR@nullem}[1]{}
\DeclareRobustCommand{\LWR@nullnormalfont}{}
```

`\LWR@nullfonts` Removes formatting during filename operations.

⚠ Use only inside a group.

The following are *not* made robust, since they must be expanded to their nullified versions.

```
\newcommand*{\LWR@nullfonts}{%
\LetLtxMacro\emph\LWR@nullemph%
\LetLtxMacro\textmd\LWR@nulltextmd%
\LetLtxMacro\textbf\LWR@nulltextbf%
\LetLtxMacro\textrm\LWR@nulltextrm%
\LetLtxMacro\textsf\LWR@nulltextsf%
\LetLtxMacro\texttt\LWR@nulltexttt%
\LetLtxMacro\textup\LWR@nulltextup%
```

```
\LetLtxMacro\textit\LWR@nulltextit%
\LetLtxMacro\textsc\LWR@nulltextsc%
\LetLtxMacro\textsl\LWR@nulltextsl%
\LetLtxMacro\textnormal\LWR@nulltextnormal%
\LetLtxMacro\rmfamily\LWR@nullrmfamily%
\LetLtxMacro\sffamily\LWR@nullsffamily%
\LetLtxMacro\ttfamily\LWR@nullttfamily%
\LetLtxMacro\bfseries\LWR@nullbfseries%
\LetLtxMacro\mdseries\LWR@nullmdseries%
\LetLtxMacro\upshape\LWR@nullupshape%
\LetLtxMacro\slshape\LWR@nullslshape%
\LetLtxMacro\scshape\LWR@nullscshape%
\LetLtxMacro\itshape\LWR@nullitshape%
\LetLtxMacro\em\LWR@nullem%
\LetLtxMacro\normalfont\LWR@nullnormalfont%


\renewcommand*{\,}{-}%
\renewcommand*{~}{-}%
\renewcommand*{\newline}{ }%
\renewcommand*{\textellipsis}{-}%

\renewcommand*{\HTMLunicode}[1]{-}%
\renewcommand*{\HTMLentity}[1]{-}%
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
\renewcommand*{\&}{and}%

\renewcommand{\textsuperscript}[1]{##1}%
\renewcommand{\textsubscript}[1]{##1}%


\renewcommand{\underline}[1]{##1}%

\RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
\DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
\DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%
```

Nullify math macros.

```
\def\(##1\){}%
\def\[##1\]{}%
\RenewDocumentCommand{\LWR@subsingledollar}{s m m m}{}%
```

Use the simpler form with `\texorpdfstring`:

```
\let\texorpdfstring\relax%
\newcommand{\texorpdfstring}[2]{##2}%
}
```

Remembers the current font family, series, and shape.

```
\newcommand*{\LWR@f@family}{rm}
\newcommand*{\LWR@f@series}{md}
\newcommand*{\LWR@f@shape}{up}
```

`\LWR@textcurrentfont` {⟨*text*⟩}

Prints the text with the current font choices.

```
\newcommand*{\LWR@textcurrentfont}[1]{%
\InlineClass{%
        text\LWR@f@family\LWR@origtilde{}%
        text\LWR@f@series\LWR@origtilde{}%
        text\LWR@f@shape%
    }%
    {#1}%
}
```

Env `LWR@blocktextcurrentfont` Prints the contents with the current font choices.

```
\newenvironment*{LWR@blocktextcurrentfont}{%
\BlockClass{%
        text\LWR@f@family\LWR@origtilde{}%
        text\LWR@f@series\LWR@origtilde{}%
        text\LWR@f@shape%
    }%
}{\endBlockClass}
```

`\mdseries`

```
\renewrobustcmd*{\mdseries}{\renewcommand*{\LWR@f@series}{md}}
```

`\bfseries`

```
\renewrobustcmd*{\bfseries}{\renewcommand*{\LWR@f@series}{bf}}
```

`\rmfamily`

```
\renewrobustcmd*{\rmfamily}{\renewcommand*{\LWR@f@family}{rm}}
```

\sffamily

```
10060 \renewrobustcmd*{\sffamily}{\renewcommand*{\LWR@f@family}{sf}}
```

\ttfamily

```
10061 \renewrobustcmd*{\ttfamily}{\renewcommand*{\LWR@f@family}{tt}}
```

\upshape

```
10062 \renewrobustcmd*{\upshape}{\renewcommand*{\LWR@f@shape}{up}}
```

\itshape

```
10063 \renewrobustcmd*{\itshape}{\renewcommand*{\LWR@f@shape}{it}}
```

\scshape

```
10064 \renewrobustcmd*{\scshape}{\renewcommand*{\LWR@f@shape}{sc}}
```

\normalfont

```
10065 \renewrobustcmd*{\normalfont}{\rmfamily\mdseries\upshape}
```

\sp {⟨*text*⟩}

For **siunitx**. Must work in math mode.

```
10066 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}{}}
```

\sb {⟨*text*⟩}

For **siunitx**. Must work in math mode.

```
10067 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}{}}
```

\textsuperscript {⟨*text*⟩}

```
10068 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

\@textsuperscript {⟨*text*⟩}

```
10069 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

\textsubscript {⟨*text*⟩}

```
\AtBeginDocument{
\renewrobustcmd{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
}
```

\@textsubscript {⟨*text*⟩}

```
\AtBeginDocument{
\renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
}
```

\up {⟨*text*⟩} Prints superscript.

This is \let at the beginning of the document in case some other package has changed the definition.

```
\AtBeginDocument{\let\up\textsuperscript}
```

\fup {⟨*text*⟩} Prints superscript.

Supports **fmtcount** package.

This is \let at the beginning of the document in case some other package has changed the definition.

```
\AtBeginDocument{\let\fup\textsuperscript}
```

\underline {⟨*text*⟩}

```
\renewcommand{\underline}[1]{%
\LWR@HTMLtextstyle%
    {text-decoration:underline; text-decoration-skip: auto}%
    {underline}{#1}%
}
```

\LWR@overline {⟨*text*⟩}

```
\newcommand{\LWR@overline}[1]{%
\LWR@HTMLtextstyle%
    {text-decoration:overline; text-decoration-skip: auto}%
    {overline}{#1}%
}
```

\LWR@currenttextcolor The color to use for text and \rule, defaulting to black:

```
\newcommand*{\LWR@currenttextcolor}{black}
```

`\LWR@tempcolor` `\LWR@tempcolortwo` The color converted to HTML colorspace.

```
\newcommand*{\LWR@tempcolor}{}
\newcommand*{\LWR@tempcolortwo}{}
```

`\LWR@findcurrenttextcolor` Sets `\LWR@tempcolor` to the current color.

```
\newcommand*{\LWR@findcurrenttextcolor}{%
\renewcommand{\LWR@tempcolor}{black}%
}
```

`\LWR@textcurrentcolor` `{`⟨*text*⟩`}` Like `\textcolor` but uses the current `\color` instead.

```
\NewDocumentCommand{\LWR@textcurrentcolor}{m}{%
    \renewcommand*{\LWR@currenttextcolor}{black}%
    #1%
}
```

```
\end{warpHTML}
```

87 Skips, spaces, font sizes

for HTML output:

```
\begin{warpHTML}
```

`\,` must be redefined after `\RequirePackage{printlen}`

Direct-formatting space commands become HTML entities:

```
\renewrobustcmd*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space
\renewrobustcmd*{\thinspace}{\HTMLunicode{202f}}    % HTML thin non-breakable space


\renewrobustcmd*{~}{\HTMLentity{nbsp}}

\renewrobustcmd*{\textellipsis}{\HTMLunicode{2026}}
```

Direct-formatting font sizes are ignored:

```
\newrobustcmd*{\LWR@HTML@normalsize}{}
\LWR@formatted{normalsize}

```

```
\newrobustcmd*{\LWR@HTML@small}{}
\LWR@formatted{small}

\newrobustcmd*{\LWR@HTML@footnotesize}{}
\LWR@formatted{footnotesize}

\newrobustcmd*{\LWR@HTML@scriptsize}{}
\LWR@formatted{scriptsize}

\newrobustcmd*{\LWR@HTML@tiny}{}
\LWR@formatted{tiny}

\newrobustcmd*{\LWR@HTML@large}{}
\LWR@formatted{large}

\newrobustcmd*{\LWR@HTML@Large}{}
\LWR@formatted{Large}

\newrobustcmd*{\LWR@HTML@LARGE}{}
\LWR@formatted{LARGE}

\newrobustcmd*{\LWR@HTML@huge}{}
\LWR@formatted{huge}

\newrobustcmd*{\LWR@HTML@Huge}{}
\LWR@formatted{Huge}


\DeclareDocumentCommand{\onecolumn}{}{}

\DeclareDocumentCommand{\twocolumn}{O{}}{

#1

}
```

\hfill

```
\newcommand*{\LWR@HTML@hfill}{\qquad}
\LWR@formatted{hfill}
```

\hrulefill

```
\newcommand*{\LWR@HTML@hrulefill}{\rule{1in}{1pt}}
\LWR@formatted{hrulefill}
```

\dotfill

```
\newcommand*{\LWR@HTML@dotfill}{\dots}
\LWR@formatted{dotfill}
```

\newpage

```
\renewcommand*{\newpage}{
}
```

\newline Uses the HTML <br /> element.

```
\newrobustcmd*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
\LetLtxMacro\newline\LWR@newlinebr
```

\\ Redefined to \LWR@endofline or \LWR@tabularendofline.

\LWR@endofline * [⟨*len*⟩]

\\ is assigned to \LWR@endofline at \LWR@LwarpStart.

Inside tabular, \\ is temporarily changed to \LWR@tabularendofline.

```
\LetLtxMacro\LWR@origendofline\\
\NewDocumentCommand{\LWR@endofline}{s o}
{%
\newline%
}
```

\LWR@minipagestartpars Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a to be inside a p, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, lwarp correctly surpresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. lwarp tries to do this by remembering that a minipage has been seen, in which case paragraph tags are surpressed around \hspace, \enskip, \quad, and \qquad until the end of the paragraph, when the closing p tag is created.

\hspace
\enskip
\quad
\qquad

When a minipage is seen, the boolean LWR@minipagethispar is set, telling the following horizontal whitespace commands to try to surpress their surrounding paragraph tags. LWR@minipagethispar is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before \hspace, \quad, or \qquad's HTML output.

```
\newcommand*{\LWR@minipagestartpars}{%
\ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
}
```

\LWR@minipagestoppars Placed just after \hspace, \quad, or \qquad's HTML output.

```
\newcommand*{\LWR@minipagestoppars}{%
\ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
}
```

\quad Handles special minipage & horizontal space interactions.

```
\renewcommand*{\quad}{%
\LWR@minipagestoppars%
\HTMLunicode{2001}%
\LWR@minipagestartpars%
}
```

\qquad Handles special minipage & horizontal space interactions.

```
\renewcommand*{\qquad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```
\renewcommand*{\enskip}{%
\LWR@minipagestoppars%
\HTMLunicode{2000}%
\LWR@minipagestartpars%
}
```

Len \LWR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

Len \LWR@tempheight

Len \LWR@tempraise

```
\newlength{\LWR@tempwidth}
\newlength{\LWR@tempheight}
\newlength{\LWR@tempraise}
```

\LWR@select@html@hspace * {⟨*length*⟩} * {⟨*length*⟩}

\hspace

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

\hspace{\fill} is converted to \hspace{2em}, equal to \qquad.

```
\newcommand{\LWR@select@html@hspace}{%
\RenewDocumentCommand{\hspace}{s m}{%
\setlength{\LWR@tempwidth}{##2}%
```

If \fill, change to \qquad:

```
\ifnum\gluestretchorder\LWR@tempwidth>0%
\setlength{\LWR@tempwidth}{2em}%
\fi%
```

Only if the width is not zero:

```
\ifdimcomp{\LWR@tempwidth}{=}{0pt}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
    \ifdimcomp{\LWR@tempwidth}{=}{.16667em}%
    {%
        \HTMLunicode{2009}% thin breakable space
    }%
```

Print the span with the converted width. Not rounded.

```
    {%
        \LWR@htmltagc{%
            span style="width:\LWR@printlength{\LWR@tempwidth};  % extra space
                display:inline-block"%
        }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```
        \ifbool{FormatWP}{%
            \setlength{\LWR@templengthone}{\LWR@tempwidth}%
            \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
                \quad%
                \addtolength{\LWR@templengthone}{-1em}%
            }%
        }{}%
```

Close the span:

```
        \LWR@htmltagc{/span}%
```

```
    }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestartpars%
}% width not 0
}%
}
```

`\LWR@select@html@nohspace` * {⟨*length*⟩}
`\hspace`

Used to disable `\hspace` while creating `description` `\items`.

```
\newcommand{\LWR@select@html@nohspace}{%
    \RenewDocumentCommand{\hspace}{s m}{}%
}
```

`\LWR@select@print@hspace`

```
\newcommand*{\LWR@select@print@hspace}{%
    \renewrobustcmd\hspace{\@ifstar\@hspacer\@hspace}%
}
```

`\hspace` * {⟨*length*⟩}

Handles special minipage & horizontal space interactions.

```
\LWR@select@html@hspace
```

`\LWR@vspace` * {⟨*length*⟩} Nullified vspace.

```
\NewDocumentCommand{\LWR@HTML@vspace}{s m}{}

\LWR@formatted{vspace}
```

`\linebreak` [⟨*num*⟩] Inserts an HTML `br` tag.

```
\renewcommand*{\linebreak}[1][]{\newline}
```

`\nolinebreak` [⟨*num*⟩]

```
\renewcommand*{\nolinebreak}[1][]{}
```

`\pagebreak` `[`⟨*num*⟩`]` Starts a new paragraph.

```
\renewcommand*{\pagebreak}[1][]{
}
```

`\nopagebreak` `[`⟨*num*⟩`]`

```
\renewcommand*{\nopagebreak}[1][]{}
```

`\enlargethispage` `*` `{`⟨*len*⟩`}`

```
\RenewDocumentCommand{\enlargethispage}{s m}{}
```

`\clearpage`
`\cleardoublepage`

```
\renewcommand*{\clearpage}{}
\renewcommand*{\cleardoublepage}{}
```

`\rule` `[`⟨*raise*⟩`]` `{`⟨*width*⟩`}` `{`⟨*height*⟩`}`

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
\newcommand*{\LWR@HTML@rule}[3][]{%
```

The width is copied into a temporary LaTeX length, from which comparisons and conversions may be made:

```
\setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
\ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
{}% zero- width
{% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
    \ifthenelse{%
        \lengthtest{\LWR@tempwidth>0pt}\AND%
        \lengthtest{\LWR@tempwidth<1pt}%
```

```
    }%
    {\setlength{\LWR@tempwidth}{1pt}}{}%
```

Likewise with height:

```
    \setlength{\LWR@tempheight}{#3}%
    \ifthenelse{%
        \lengthtest{\LWR@tempheight>0pt}\AND%
        \lengthtest{\LWR@tempheight<1pt}%
    }%
    {\setlength{\LWR@tempheight}{1pt}}{}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
    \LWR@minipagestoppars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in LaTeX code.

```
    \LWR@findcurrenttextcolor%
    \LWR@htmltagc{%
    span
    style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
    \ifbool{FormatWP}{}{background:\LWR@currenttextcolor ; }%
```

The width and height are printed, converted to PT:

```
    width:\LWR@printlength{\LWR@tempwidth} ; %
    height:\LWR@printlength{\LWR@tempheight} ; %
```

The raise height is converted to a CSS transform. The `*2` raise multiplier is to approximately match HTML output's X height. Conversion to a LaTeX length allows a typical LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of LaTeX ex height, then print that to HTML with an ex unit.

```
    \ifblank{#1}%
    {}%
    {%
        \setlength{\LWR@tempraise}{0pt-#1}%
        \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
```

```
        \LWR@orignewline%
        -ms-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
        \LWR@orignewline%
        -webkit-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
        \LWR@orignewline%
        transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
        \LWR@orignewline%
    }%
```

Display inline-block to place the span inline with the text:

```
    display:inline-block;"%
    }%
```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```
        \ifbool{FormatWP}{%
            \setlength{\LWR@templengthone}{\LWR@tempwidth}%
            \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
                \_{}%
                \addtolength{\LWR@templengthone}{-1em}%
            }%
        }{}%
```

Close the span:

```
    \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
    \LWR@minipagestartpars%
}% non-zero width
}

\LWR@formatted{rule}

\end{warpHTML}
```

88 \phantomsection

for HTML output:

```
\begin{warpHTML}
```

\phantomsection Emulate the **hyperref** \phantomsection command, often used to insert the bibliography into the table of contents. Ignores \ForceHTMLTOC.

```
\DeclareDocumentCommand{\phantomsection}{}{%
\begingroup%
\boolfalse{LWR@forcinghtmltoc}%
\section*{}%
\endgroup%
}
```

```
\end{warpHTML}
```

89 \LaTeX and other logos

Logos for HTML and print modes:

Some of these logos may be redefined in a later package, so after loading other packages, and at the beginning of the document, their definitions are finally \let in \LWR@LwarpStart.

For CSS conversions, see:
http://edward.oconnor.cx/2007/08/tex-poshlet
http://nitens.org/taraborelli/texlogo

89.1 HTML logos

for HTML output:

```
\begin{warpHTML}
```

\TeX TEX

latexlogo is a CSS class used to properly typeset the E and A in LATEX and friends.

latexlogofont is a CSS class used to select the font for the rest of the logo in LATEX, LuaTEX, ConTEXt, etc.

```
\let\LWR@origTeX\TeX

\newcommand*{\LWR@TeX}
{%
    \InlineClass{latexlogofont}%
    {%
        \LWR@HTMLtextstyle%
            {text-transform:uppercase}%
```

```
            {latexlogo}%
        {T\textsubscript{e}X}%
    }%
}
```

\LaTeX LaTeX, LaTeX 2ε
\LaTeXe

```
\let\LWR@origLaTeX\LaTeX

\newcommand*{\LWR@LaTeX}
{%
    \InlineClass{latexlogofont}%
    {%
        \LWR@HTMLtextstyle%
            {text-transform:uppercase}%
            {latexlogo}%
        {L\textsuperscript{a}T\textsubscript{e}X}%
    }%
}

\let\LWR@origLaTeXe\LaTeXe

\renewcommand*{\LaTeXe}
{\LaTeX\InlineClass{latexlogofont}%
{\,2\textsubscript{\textit{\HTMLunicode{3B5}}}}}
```

\LuaTeX LuaTeX, LuaLaTeX
\LuaLaTeX

```
\newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
\newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

\XeTeX XeTeX, XeLaTeX
\XeLaTeX

xetexlogo is a css class which aligns the backwards E in XeTeX and spaces TeX appropriately.

xelatexlogo is a css class which aligns the backwards E in XeLaTeX and spaces LaTeX appropriately.

```
\newcommand*{\Xe}
  {X\textsubscript{\HTMLunicode{18e}}}
\newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
\newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

\ConTeXt ConTeXt

```
\newcommand*{\LWR@ConTeXt}
{\InlineClass{latexlogofont}{Con}\TeX{}%
\InlineClass{latexlogofont}{t}}
```

`\BibTeX` `\MakeIndex` BibTeX, *MakeIndex*

```
\providecommand*{\BibTeX}
{\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}

\newcommand*{\MakeIndex}
{\InlineClass{latexlogofont}{\textit{MakeIndex}}}
```

`\AmS` $\mathcal{AMS}$

amslogo is a css class used for the $\mathcal{AMS}$logo.

```
\AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
{\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

`\MiKTeX` MiKTeX

```
\newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` LyX

lyxlogo is a css class used for the LyXlogo.

```
\newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
\end{warpHTML}
```

89.2 Print logos

for PRINT output:

```
\begin{warpprint}
\newcommand*{\XeTeXrevE}
  {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
\providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
\providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
\providecommand*{\AmS}{%
\leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
\hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
\newcommand*{\LyX}{\textsf{LyX}}
\providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
\providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
```

```
\providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
\providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
\providecommand*{\ConTeXt}{\mbox{Con\TeX{}t}}
\providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
\end{warpprint}
```

90 \AtBeginDocument, \AtEndDocument

for HTML output:

```
\begin{warpHTML}
```

\LWR@LwarpStart
\LWR@LwarpEnd

Automatically sets up the HTML-related actions for the start and end of the document.

```
\AfterEndPreamble{\LWR@LwarpStart}
\AtEndDocument{\LWR@LwarpEnd}

\end{warpHTML}
```

91 Koma-script

Load patches to **koma-script**.

for HTML output:

```
\begin{warpHTML}

\@ifclassloaded{scrbook}{\RequirePackage{lwarp-patch-komascript}}{}
\@ifclassloaded{scrartcl}{\RequirePackage{lwarp-patch-komascript}}{}
\@ifclassloaded{scrreprt}{\RequirePackage{lwarp-patch-komascript}}{}

\end{warpHTML}
```

92 Memoir

Load patches to **memoir**.

for HTML output:

```
\begin{warpHTML}

\@ifclassloaded{memoir}{\RequirePackage{lwarp-patch-memoir}}{}

\end{warpHTML}
```

93 Trademarks

- TEX is a trademark of American Mathematical Society.
- ADOBE® and ADOBE **Framemaker**® are either registered trademarks or trademarks of ADOBE SYSTEMS INCORPORATED in the United States and/or other countries.
- LINUX® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- MAC OS® is a trademark of APPLE INC.
- MADCAP FLARE™ is the property of MADCAP SOFTWARE, INC.
- MATHJAX is copyright 2009 and later. The MATHJAX CONSORTIUM is a joint venture of the AMERICAN MATHEMATICAL SOCIETY (AMS) and the SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS (SIAM) to advance mathematical and scientific content on the web.
- MICROSOFT®, ENCARTA, MSN, and WINDOWS® are either registered trademarks or trademarks of MICROSOFT CORPORATION in the United States and/or other countries.
- UNIX® is a registered trademark of THE OPEN GROUP.

File 2 `lwarp-2up.sty`

§ 94 Package 2up

Pkg 2up **2up** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{2up}[2010/05/15]

\def\source#1#2#3{}
\def\target#1#2#3{}
\def\targetlayout#1{}
\newdimen\pageseplength
\newdimen\pagesepwidth
\newdimen\pagesepoffset
\def\twoupemptypage{}
\def\twoupclearpage{}
\def\twoupeject{}
\def\twouparticle{}
\def\twoupplain{}
\def\twouplegaltarget{}
\def\twouplandscape{}
\def\TwoupWrites{}
```

File 3 `lwarp-a4.sty`

§ 95 Package a4

Pkg a4 **a4** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{a4}

\newcommand*{\WideMargins}{}
```

File 4 `lwarp-a4wide.sty`

§ 96 Package a4wide

Pkg a4wide **a4wide** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{a4wide}
```

File 5 `lwarp-a5comb.sty`

§ 97 Package a5comb

Pkg a5comb **a5comb** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{a5comb}
```

File 6 `lwarp-abstract.sty`

§ 98 Package abstract

(*Emulates or patches code by* Peter Wilson.)

Pkg abstract **abstract** is supported and patched by **lwarp**.

⚠ missing TOC If using the `number` option with file splits, be sure to place the table of contents before the abstract. The `number` option causes a section break which may cause a file split, which would put a table of contents out of the home page if it is after the abstract.

for HTML output: **memoir** provides an `abstract` environment even though it is not an **article** or **report** class. Meanwhile, **lwarp** loads **book** to emulate **memoir**, but **book** does not have an `abstract` environment, so when the **abstract** package is loaded for emulation there is no pre-existing `abstract` to redefine, which would cause an error. Thus, a null `abstract` is provide here:

```
\ProvideDocumentEnvironment{abstract}{}{}{}
```

Accept all options for **lwarp-abstract**:

```
\LWR@ProvidesPackagePass{abstract}

\AtBeginDocument{
\BeforeBeginEnvironment{abstract}{
\LWR@forcenewpage
\BlockClass{abstract}
}
\AfterEndEnvironment{abstract}{\endBlockClass}
}

\renewcommand{\@bsrunintitle}{%
```

```
\hspace*{\abstitleskip}%
{\abstractnamefont%
\InlineClass{abstractrunintitle}{\abstractname}%
\@bslabeldelim}%
}

\@ifclassloaded{memoir}
{
  \renewenvironment{abstract}{%
%     \titlepage
    \null\vfil
    \@beginparpenalty\@lowpenalty
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
          \begin{\absnamepos}%
  \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
            \@endparpenalty\@M
          \end\absnamepos%
%%        \vspace{\abstitleskip}%
        \fi
      \fi
      \vspace{\abstitleskip}%
    \fi
    \put@bsintoc%
    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
    }
}{% not memoir
\if@titlepage
  \renewenvironment{abstract}{%
%     \titlepage
    \null\vfil
    \@beginparpenalty\@lowpenalty
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
          \begin{\absnamepos}%
  \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
            \@endparpenalty\@M
```

```
          \end\absnamepos%
%%        \vspace{\abstitleskip}%
        \fi
      \fi
      \vspace{\abstitleskip}%
    \fi
    \put@bsintoc%
    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
    }
\else
  \renewenvironment{abstract}{%
    \if@bsrunin
    \else
      \if@bsstyle
        \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
      \else
        \ifnumber@bs
          \num@bs
        \else
\begin{\absnamepos}%
\abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
\end\absnamepos%
%%        \vspace{\abstitleskip}%
        \fi
      \fi
      \vspace{\abstitleskip}%
    \fi
    \put@bsintoc%
    \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
    {\par\end{@bstr@ctlist}}
\fi
}% not memoir
```

File 7 `lwarp-accsupp.sty`

§ 99 Package accsupp

Pkg `accsupp` **accsupp** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{accsupp}

\newcommand*{\BeginAccSupp}[1]{}
\newcommand*{\EndAccSupp}[1]{}
```

File 8 lwarp-acro.sty

§ 100 Package acro

(*Emulates or patches code by* Clemens Niederberger.)

Pkg acro **acro** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{acro}
```

\DeclareAcronym is used in the preamble, where **lwarp** has not yet made the dollar active, so temporarily enable **lwarp** math catcode just for this definition:

```
\ExplSyntaxOn
\NewDocumentCommand \LWR@DeclareAcronym {mm}
{
    \acro_declare_acronym:nn {#1} {#2}
    \catcode`\$=3% lwarp
}
\ExplSyntaxOff

\RenewDocumentCommand{\DeclareAcronym}{}{
    \catcode`\$=\active% lwarp
    \LWR@DeclareAcronym
}
```

Modified to activate the current font:

```
\ExplSyntaxOn
\cs_gset_protected:Npn \acro_write_short:nn #1#2
  {
    \mode_if_horizontal:F { \leavevmode }
    \group_begin:
      \bool_if:NTF \l__acro_custom_format_bool
        { \l__acro_custom_format_tl }
        { \l__acro_short_format_tl }
      {\LWR@textcurrentfont{#2}}% lwarp
    \group_end:
  }

\cs_gset_protected:Npn \acro_write_alt:nn #1#2
  {
    \mode_if_horizontal:F { \leavevmode }
    \group_begin:
      \bool_if:NTF \l__acro_custom_format_bool
```

```
        { \l__acro_custom_format_tl }
        { \l__acro_alt_format_tl }
      {\LWR@textcurrentfont{#2}}% lwarp
    \group_end:
  }

\cs_gset_protected:Npn \acro_write_long:nn #1#2
  {
    \mode_if_horizontal:F { \leavevmode }
    \group_begin:
      \bool_if:NTF \l__acro_custom_long_format_bool
        { \l__acro_custom_long_format_tl }
        { \use:n }
      {
        \use:x
          {
            \exp_not:n {#1}
            {
              \bool_if:NTF \l__acro_first_upper_bool
                { \exp_not:N \__acro_first_upper_case:n { \exp_not:n {
                    \LWR@textcurrentfont{#2}% lwarp
                } } }
                { \exp_not:n {\LWR@textcurrentfont{#2}} }% lwarp
            }
          }
      }
    \group_end:
  }
\ExplSyntaxOff
```

File 9 `lwarp-acronym.sty`

§ 101 Package acronym

(*Emulates or patches code by* Tobias Oetiker.)

Pkg `acronym` **acronym** is patched for use by **lwarp**.

⚠ `\acresetall` does not work with **cleveref**, causing multiply-defined labels. **lwarp** patches **acronym** for HTML, but not for print mode.

for HTML output:

```
\LWR@ProvidesPackagePass{acronym}
```

Uses `\textit` instead of `\itshape`:

```
\renewcommand{\acfia}[1]{%
  {\textit{\AC@acl{#1}}} (\ifAC@starred\acs*{#1}\else\acs{#1}\fi)}
```

Removes the mbox to allow math inside:

```
\renewcommand*\AC@acs[1]{%
%     \mbox{
 \expandafter\AC@get\csname fn@#1\endcsname\@firstoftwo{#1}}
% }
```

Modified for **cleveref** and **zref**:

```
\renewcommand*\AC@und@newl@bel[3]{%
    \@ifundefined{#1@#3}%
    {%
        \global\expandafter\let\csname#2@#3\endcsname\@nnil
        \global\expandafter\let\csname#2@#3@cref\endcsname\@nnil% lwarp
    }%
    {%
        \global\expandafter\let\csname#1@#3\endcsname\relax
        \global\expandafter\let\csname#1@#3@cref\endcsname\relax% lwarp
        \global\expandafter\let\csname Z@R@#3\endcsname\relax% lwarp
    }%
}%
```

Modified for **cleveref** and **zref**:

```
\renewcommand*\AC@testdef[3]{%
\ifstrequal{#1}{Z@R}{}{% lwarp
  \@ifundefined{s@#2}\@secondoftwo\@firstofone
  {%
    \expandafter\ifx\csname s@#2\endcsname\empty
      \expandafter\@firstofone
    \else
      \expandafter\xdef\csname s@#2\endcsname{%
        \expandafter\expandafter
        \expandafter\@gobble
        \csname s@#2\endcsname
      }%
      \expandafter\@gobble
    \fi
  }%
  {%
    \@testdef{#1}{#2}{#3}%
  }%
}% lwarp
}%
```

File 10 `lwarp-adjmulticol.sty`

§ 102 Package adjmulticol

(*Emulates or patches code by* Boris Veytsman.)

Pkg `adjmulticol` **adjmulticol** is emulated.

Emulation similar to **multicols** is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, **lwarp** allows a variable number of columns up to three.

for HTML output:

```
\LWR@ProvidesPackageDrop{adjmulticol}
\RequirePackage{multicol}
```

`adjmulticols` * {⟨*numcols*⟩} {⟨*left margi*⟩} {⟨*right margin*⟩}

```
\NewDocumentEnvironment{adjmulticols}{s m m m}
{%
```

Compute the margins, and limit to positive only:

```
\setlength{\LWR@templengthone}{#3}%
\ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{}%
\setlength{\LWR@templengthtwo}{#4}
\ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{}%
```

If one column is specified, use a `<div>` of class `singlecolumn`, else use `multicols`:

```
\newcommand*{\LWR@mcolstype}{multicols}%
\ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mcolstype}{singlecolumn}}{}%
```

Help avoid page overflow:

```
\LWR@forcenewpage%
```

Create the `<div>` with the given margin and class:

```
\BlockClass[%
\LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
\LWR@print@mbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}%
]{\LWR@mcolstype}%
}
```

```
17 {\endBlockClass}
```

File 11 `lwarp-addlines.sty`

§ 103 Package addlines

(*Emulates or patches code by* Will Robertson.)

Pkg addlines **addlines** is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{addlines}

2 \newcommand*\addlines[1][1]{}
3 \let\addline\addlines
4 \newcommand*\removelines[1][1]{}
5 \let\removeline\removelines
```

File 12 `lwarp-ae.sty`

§ 104 Package ae

Pkg ae **ae** does not work with **pdftotext**, and is superceded by **latinmodern**.

for HTML output:

```
1 \LWR@loadnever{ae}{latinmodern}
```

File 13 `lwarp-aecc.sty`

§ 105 Package aecc

Pkg aecc **aecc** does not work with **pdftotext**, and is superceded by **latinmodern**.

for HTML output:

```
1 \LWR@loadnever{aecc}{latinmodern}
```

File 14 `lwarp-afterpage.sty`

§ 106 Package afterpage

(*Emulates or patches code by* David Carlisle.)

Pkg afterpage Emulated.

for HTML output: Discard all options for **lwarp-afterpage**:

```
\LWR@ProvidesPackageDrop{afterpage}
\newcommand{\afterpage}[1]{#1}
```

File 15 `lwarp-algorithm2e.sty`

§ 107 Package algorithm2e

(*Emulates or patches code by* Christophe Fiorio.)

Pkg algorithm2e **algorithm2e** is patched for use by **lwarp**.

For print output, captions are placed according to package options, but for HTML output captions are placed where used. Therefore, to have captions appear at the top of the algorithms for both print and HTML, place each captions at the top of each algorithm.

for HTML output:

```
\LWR@ProvidesPackagePass{algorithm2e}
```

For the list-of entries:

```
\renewcommand{\l@algocf}[2]{\hypertocfloat{1}{algocf}{loa}{#1}{#2}}
```

Select the **lwarp** float style according to the **algorithm2e** style:

```
\newcommand*{\LWR@floatstyle@algocf}{ruled}

\ifdefstring{\algocf@style}{boxed}{%
\renewcommand*{\LWR@floatstyle@algocf}{boxed}
}{}

\ifdefstring{\algocf@style}{boxruled}{%
\renewcommand*{\LWR@floatstyle@algocf}{boxruled}
}{}

\ifdefstring{\algocf@style}{plain}{%
\renewcommand*{\LWR@floatstyle@algocf}{plain}
}{}
```

Paragraph handling to allow line numbers under certain conditions:

```
\newbool{LWR@algocf@dopars}
```

```
\booltrue{LWR@algocf@dopars}

\renewcommand{\algocf@everypar}{%
\ifbool{LWR@algocf@dopars}{%
    \ifbool{LWR@doingstartpars}{%
        \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
            {}%
            {%
                \algocf@everyparnl\algocf@everyparhanging%
            }%
    }{}%
}{}%
}
```

lwarp caption handling:

```
\renewcommand{\algocf@makecaption}[2]{%
\LWR@HTML@caption@begin{algocf}%
\algocf@captiontext{#1}{#2}%
\LWR@HTML@caption@end%
}
```

Print any caption where it is declared:

```
\renewcommand{\algocf@makecaption@plain}[2]{%
    \LWR@HTML@caption@begin{algocf}%
    \algocf@captiontext{#1}{#2}%
    \LWR@HTML@caption@end%
}

\renewcommand{\algocf@makecaption@boxed}[2]{%
    \LWR@HTML@caption@begin{algocf}%
    \algocf@captiontext{#1}{#2}%
    \LWR@HTML@caption@end%
}

\renewcommand{\algocf@makecaption@ruled}[2]{%
    \LWR@HTML@caption@begin{algocf}%
    \algocf@captiontext{#1}{#2}%
    \LWR@HTML@caption@end%
}
```

Turn off line numbering while making the caption:

```
\long\def\algocf@latexcaption#1[#2]#3{% original definition of caption
\boolfalse{LWR@algocf@dopars}%  lwarp
  \par%
  \addcontentsline{\csname ext@#1\endcsname}{#1}%
  {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
```

```
  \begingroup%
  \@parboxrestore%
  \if@minipage%
    \@setminipage%
  \fi%
  \normalsize%
  \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par%
  \endgroup%
\booltrue{LWR@algocf@dopars}%   lwarp
}
```

Line numbers are printed in a of class `alg2elinenumber`:

```
\renewcommand{\algocf@printnl}[1]{%
    \InlineClass{alg2elinenumber}{\NlSty{#1}}~%
}%
```

While initializing an algorithm environment, locally declare the style of a regular figure to be the same as the algorithm style, in case the `figure` option was used.

```
\preto\@algocf@init{%
  \edef\LWR@floatstyle@figure{\LWR@floatstyle@algocf}%
}
```

For **lwarp**, the algorithm is not assembled inside a box, since `lateximage`s would not work, so the captions are printed where declared.

```
\renewcommand{\@algocf@start}{%
    \let\@mathsemicolon=\;\def\;{\ifmmode\@mathsemicolon\else\@endalgoln\fi}%
%     \raggedright%
    \AlFnt{}%
    \booltrue{LWR@algocf@dopars}% lwarp
}

\renewcommand{\@algocf@finish}{%
    \boolfalse{LWR@algocf@dopars}% lwarp
    \lineskip\normallineskip\setlength{\skiptotal}{\@defaultskiptotal}%
    \let\;=\@mathsemicolon%
    \let\]=\@emathdisplay%
}
```

Use an HTML break:

```
\renewcommand{\BlankLine}{%
\LWR@stoppars%
\LWR@htmltagc{br /}%
\LWR@startpars%
}
```

Simplified for HTML. The paragraph handling must be preserved.

```
\renewcommand{\SetKwInOut}[2]{%
  \algocf@newcommand{#1}[1]{%
    \ifthenelse{\boolean{algocf@hanginginout}}%
        {\relax}%
        {\algocf@seteveryparhanging{\relax}}%
    \ifthenelse{\boolean{algocf@inoutnumbered}}%
        {\relax}%
        {\algocf@seteveryparnl{\relax}}%
    {%
             \KwSty{#2\algocf@typo:}%
        ~##1\par%
    }%
    \algocf@linesnumbered% reset the numbering of the lines
    \ifthenelse{\boolean{algocf@hanginginout}}%
        {\relax}%
        {\algocf@reseteveryparhanging}%
  }%
}%

\renewcommand{\ResetInOut}[1]{}%
```

Each of the following creates a of a given class, and turns off line numbering while creating the tags:

```
\renewcommand{\algocf@Vline}[1]{%
\boolfalse{LWR@algocf@dopars}%
\begin{BlockClass}{alg2evline}
\booltrue{LWR@algocf@dopars}%
#1
\boolfalse{LWR@algocf@dopars}%
\end{BlockClass}
\booltrue{LWR@algocf@dopars}%
}

\renewcommand{\algocf@Vsline}[1]{%
\boolfalse{LWR@algocf@dopars}%
\begin{BlockClass}{alg2evsline}
\booltrue{LWR@algocf@dopars}%
#1
\boolfalse{LWR@algocf@dopars}%
\end{BlockClass}
\booltrue{LWR@algocf@dopars}%
}

\renewcommand{\algocf@Noline}[1]{%
\boolfalse{LWR@algocf@dopars}%
\begin{BlockClass}{alg2enoline}
```

```
\booltrue{LWR@algocf@dopars}%
#1
\boolfalse{LWR@algocf@dopars}%
\end{BlockClass}
\booltrue{LWR@algocf@dopars}%
}
```

The [H] environment is converted to a regular float, which in HTML is placed where declared. Reusing the regular float allows the [H] version to reuse the ruled and boxed options.

```
\LetLtxMacro\algocf@Here\algocf
\LetLtxMacro\endalgocf@Here\endalgocf
```

File 16 `lwarp-algorithmicx.sty`

§ 108 Package algorithmicx

(*Emulates or patches code by* Szász János.)

Pkg algorithmicx **algorithmicx** is supported with minor adjustments.

for HTML output:

```
\LWR@ProvidesPackagePass{algorithmicx}
```

Inside the algorithmic environment, level indenting is converted to a of the required length, and comments are placed inside a which is floated right.

⚠ package conflicts If using \newfloat, **trivfloat**, and/or **algorithmicx** together, see section 390.1.

for HTML output:

```
\begin{warpHTML}

\AtBeginEnvironment{algorithmic}{%
%
\let\origALG@doentity\ALG@doentity%
%
\renewcommand*{\ALG@doentity}{%
\origALG@doentity%
\LWR@htmltagc{%
span style="width:\LWR@printlength{\ALG@thistlm}; display:inline-block;"%
}%
\ifbool{FormatWP}{%
\setlength{\LWR@templengthone}{\the\ALG@thistlm}%
\whiledo{\lengthtest{\LWR@templengthone>1em}}{%
\quad%
\addtolength{\LWR@templengthone}{-1em}%
}%
```

```
}{}%
\LWR@htmltagc{/span}%
}%

\let\LWR@origComment\Comment%

\renewcommand{\Comment}[1]{%
    \InlineClass{floatright}{\LWR@origComment{#1}}%
}%
}

\renewcommand\algorithmiccomment[1]{%
\hfill\HTMLunicode{25B7} #1% white right triangle
}%

\end{warpHTML}
```

File 17 `lwarp-alltt.sty`

§ 109 Package alltt

(*Emulates or patches code by* Johannes Braams.)

Pkg `alltt` **alltt** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{alltt}

\AfterEndPreamble{
\LWR@traceinfo{Patching alltt.}
\AtBeginEnvironment{alltt}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{alltt}%
}
\AfterEndEnvironment{alltt}{%
    \LWR@afterendverbatim{2}%
}
}
```

File 18 `lwarp-amsmath.sty`

§ 110 Package amsmath

(*Emulates or patches code by* American Mathematical Society, LaTeX3 Project.)

Pkg amsmath **amsmath** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{amsmath}
```

Patches to allow \eqref inside a caption:

```
\def\maketag@@@#1{\text{#1}}
\def\tagform@#1{\maketag@@@{(\ignorespaces#1\unskip)}}
```

Patches for $\mathcal{AMS}$ math \tag macro to remember the first tag:

```
\ifbool{mathjax}{}{% not mathjax

\LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
\LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@

\renewcommand*{\make@df@tag@@}[1]{%
\LWR@remembertag{#1}%
\LWR@origmake@df@tag@@{#1}%
}

\renewcommand*{\make@df@tag@@@}[1]{%
\LWR@remembertag{#1}%
\LWR@origmake@df@tag@@@{#1}%
}

}% not mathjax
```

The following $\mathcal{AMS}$ environments are more easily patched in-place:

Env multline

```
\BeforeBeginEnvironment{multline}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \booltrue{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbodynumbered{multline}]
}
}

```

```
\AfterEndEnvironment{multline}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \boolfalse{LWR@amsmultline}
    \LWR@addmathjax{multline}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}
```

Env `multline*`

```
\BeforeBeginEnvironment{multline*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \booltrue{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbody{multline*}]
}
}

\AfterEndEnvironment{multline*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \boolfalse{LWR@amsmultline}
    \LWR@addmathjax{multline*}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}

```

Env `gather`

```
\BeforeBeginEnvironment{gather}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

```
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbodynumbered{gather}]
}
}

\AfterEndEnvironment{gather}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{gather}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}
```

Env `gather*`

```
\BeforeBeginEnvironment{gather*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbody{gather*}]
}
}

\AfterEndEnvironment{gather*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{gather*}{\the\@envbody}
}
```

```
{\end{lateximage}\end{BlockClass}}

}
```

Env `align`

```
\BeforeBeginEnvironment{align}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
}
}

\AfterEndEnvironment{align}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{align}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}
```

Env `align*`

```
\BeforeBeginEnvironment{align*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbody{align*}]
```

```
}
}

\AfterEndEnvironment{align*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{align*}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}
```

Env `flalign`

```
\BeforeBeginEnvironment{flalign}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
}
}

\AfterEndEnvironment{flalign}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{flalign}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}
```

Env `flalign*`

```
\BeforeBeginEnvironment{flalign*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

```
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbody{flalign*}]
}
}

\AfterEndEnvironment{flalign*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{flalign*}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}
```

Env `alignat`

```
\BeforeBeginEnvironment{alignat}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbodynumbered{alignat}]
}
}

\AfterEndEnvironment{alignat}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{alignat}{\the\@envbody}
}
```

```
{\end{lateximage}\end{BlockClass}}

}
```

Env `alignat*`

```
\BeforeBeginEnvironment{alignat*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@syncmathjax
    \boolfalse{LWR@amsmultline}
    \LWR@beginhideamsmath
}
{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[\LWR@amsmathbody{alignat*}]
}
}

\AfterEndEnvironment{alignat*}{

\ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
{
    \LWR@endhideamsmath
    \LWR@addmathjax{alignat*}{\the\@envbody}
}
{\end{lateximage}\end{BlockClass}}

}
```

File 19 `lwarp-amsthm.sty`

§ 111 Package amsthm

(*Emulates or patches code by* Publications Technical Group — American Mathematical Society.)

The original source code is located in `amsclass.dtx`, and printed in `amsclass.pdf`.

Pkg `amsthm` **amsthm** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{amsthm}
```

Table 12: AMSthm package — CSS styling of theorems and proofs

Theorem: of class amsthmbody<theoremstyle>

Theorem Name: of class amsthmname<theoremtyle>

Theorem Number: of class amsthmnumber<theoremstyle>

Theorem Note: of class amsthmnote<theoremstyle>

Proof: of class amsthmproof

Proof Name: of class amsthmproofname

where <theoremstyle> is plain, definition, etc.

Storage for the style being used for new theorems:

```
\newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
\renewcommand{\theoremstyle}[1]{%
  \@ifundefined{th@#1}{%
    \PackageWarning{amsthm}{Unknown theoremstyle `#1'}%
    \thm@style{plain}%
    \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
  }{%
    \thm@style{#1}%
    \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
  }%
}
```

Patched to remember the style for this theorem type:

```
\def\@xnthm#1#2{%
  \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% lwarp
  \let\@tempa\relax
  \@xp\@ifdefinable\csname #2\endcsname{%
    \global\@xp\let\csname end#2\endcsname\@endtheorem
    \ifx *#1% unnumbered, need to get one more mandatory arg
      \edef\@tempa##1{%
        \gdef\@xp\@nx\csname#2\endcsname{%
          \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
            {}{##1}}}%
    \else % numbered theorem, need to check for optional arg
      \def\@tempa{\@oparg{\@ynthm{#2}}[]}%
    \fi
```

```
    \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#2}}}% lwarp
  }%
  \@tempa
}
```

Patched to enclose with css:

```
\newcommand{\LWR@haveamsthmname}{
\renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
}

\newcommand{\LWR@haveamsthmnumber}{
\renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
}

\newcommand{\LWR@haveamsthmnote}{
\renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
}

\LWR@haveamsthmname
\LWR@haveamsthmnumber
\LWR@haveamsthmnote
```

Patches for css:

```
\def\@begintheorem#1#2[#3]{%
    \item[
%   \deferred@thm@head{
%     \the\thm@headfont \thm@indent
    \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% lwarp
    \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% lwarp
    \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% lwarp
    \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
    \the\thm@headpunct~
    \thmheadnl % possibly a newline.
    \hskip\thm@headsep
%   }%
    ]
  \ignorespaces}
```

Patched for css:

```
\def\@thm#1#2#3{%
  \ifhmode\unskip\unskip\par\fi
  \normalfont
  \LWR@forcenewpage% lwarp
  \BlockClass{amsthmbody\LWR@thisthmstyle}% lwarp
  \trivlist
  \let\thmheadnl\relax
```

```
  \let\thm@swap\@gobble
  \thm@notefont{\fontseries\mddefault\upshape}%
  \thm@headpunct{.}% add period after heading
  \thm@headsep 5\p@ plus\p@ minus\p@\relax
  \thm@space@setup
  #1% style overrides
  \@topsep \thm@preskip               % used by thm head
  \@topsepadd \thm@postskip           % used by \@endparenv
  \def\@tempa{#2}\ifx\@empty\@tempa
    \def\@tempa{\@oparg{\@begintheorem{#3}{}}[]}%
  \else
    \refstepcounter{#2}%
    \def\@tempa{\@oparg{\@begintheorem{#3}{\csname the#2\endcsname}}[]}%
  \fi
  \@tempa
}
```

cleveref patches `\@thm` to do `\cref@thmoptarg` if an optional argument is given. **lwarp** then patches `\cref@thmoptarg` `\AtBeginDocument`.

```
\AtBeginDocument{
\def\cref@thmoptarg[#1]#2#3#4{%
    \ifhmode\unskip\unskip\par\fi%
    \normalfont%
    \LWR@forcenewpage% lwarp
    \BlockClass{amsthmbody\LWR@thisthmstyle}% lwarp
    \trivlist%
    \let\thmheadnl\relax%
    \let\thm@swap\@gobble%
    \thm@notefont{\fontseries\mddefault\upshape}%
    \thm@headpunct{.}% add period after heading
    \thm@headsep 5\p@ plus\p@ minus\p@\relax%
    \thm@space@setup%
    #2% style overrides
    \@topsep \thm@preskip               % used by thm head
    \@topsepadd \thm@postskip           % used by \@endparenv
    \def\@tempa{#3}\ifx\@empty\@tempa%
        \def\@tempa{\@oparg{\@begintheorem{#4}{}}[]}%
    \else%
        \refstepcounter[#1]{#3}%  <<< cleveref modification
        \def\@tempa{\@oparg{\@begintheorem{#4}{\csname the#3\endcsname}}[]}%
    \fi%
    \@tempa
}%
}% AtBeginDocument

\def\@endtheorem{\endtrivlist\endBlockClass\@endpefalse }
```

Proof QED symbol:

```
\AtBeginDocument{
\@ifundefined{LWR@orig@openbox}{
\LetLtxMacro\LWR@orig@openbox\openbox
\LetLtxMacro\LWR@orig@blacksquare\blacksquare
\LetLtxMacro\LWR@orig@Box\Box

\def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
\def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
\def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\openbox\LWR@orig@openbox%
\LetLtxMacro\blacksquare\LWR@orig@blacksquare%
\LetLtxMacro\Box\LWR@orig@Box%
}% appto
}{}% @ifundefined
}% AtBeginDocument
```

Patched for css:

```
\renewenvironment{proof}[1][\proofname]{\par
\LWR@forcenewpage% lwarp
    \BlockClass{amsthmproof}% lwarp
  \pushQED{\qed}%
  \normalfont \topsep6\p@\@plus6\p@\relax
  \trivlist
  \item[
        \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
}{%
  \InlineClass{theoremendmark}{\popQED}\endtrivlist%
  \endBlockClass% lwarp
  \@endpefalse
}
```

File 20 **lwarp-anonchap.sty**

§ 112 Package

anonchap

(*Emulates or patches code by* Peter Wilson.)

Pkg anonchap **anonchap** is emulated.

Pkg tocloft

⚠ **tocloft** & other packages

If using **tocloft** with **tocbibind**, **anonchap**, **fncychap**, or other packages which change chapter title formatting, load **tocloft** with its `titles` option, which tells **tocloft** to

use standard LaTeX commands to create the titles, allowing other packages to work with it.

The code is shared by **tocbibind**.

for HTML output:

```
\LWR@ProvidesPackageDrop{anonchap}

\newcommand{\simplechapter}[1][\@empty]{%
    \def\@chapcntformat##1{%
        #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
    }%
}

\newcommand{\restorechapter}{%
\let\@chapcntformat\@seccntformat%
}
```

File 21 `lwarp-anysize.sty`

§ 113 Package anysize

(*Emulates or patches code by* Michael Salzenberg, Thomas Esser.)

Pkg `anysize` **anysize** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{anysize}

\def\papersize#1#2{}
\def\marginsize#1#2#3#4{}
```

File 22 `lwarp-appendix.sty`

§ 114 Package appendix

(*Emulates or patches code by* Peter Wilson.)

Pkg `appendix` **appendix** is patched for use by **lwarp**.

⚠ incorrect TOC link During HTML conversion, the option `toc` without the option `page` results in a TOC link to whichever section was before the `appendices` environment. It is recommended to use both `toc` and also `page` at the same time.

for HTML output:

```
\LWR@ProvidesPackagePass{appendix}
```

```
\renewcommand*{\@chap@pppage}{%
\part*{\appendixpagename}
 \if@dotoc@pp
 \addappheadtotoc
 \fi
}

\renewcommand*{\@sec@pppage}{%
\part*{\appendixpagename}
 \if@dotoc@pp
 \addappheadtotoc
 \fi
}
```

File 23 `lwarp-arabicfront.sty`

§ 115 Package arabicfront

Pkg `arabicfront` **arabicfront** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{arabicfront}
```

File 24 `lwarp-array.sty`

§ 116 Package array

Pkg `array` **array** is used as-is for print output, and emulated for HTML.

for HTML output: Remove the dummy macros:

```
\let\firsthline\relax
\let\lasthline\relax

\LWR@ProvidesPackagePass{array}

\newcommand*{\LWR@HTML@firsthline}{\LWR@HTMLhline}%
\LWR@expandableformatted{firsthline}

\newcommand*{\LWR@HTML@lasthline}{\LWR@HTMLhline}%
\LWR@expandableformatted{lasthline}
```

File 25 `lwarp-arydshln.sty`

§ 117 Package

arydshln

(*Emulates or patches code by* Hiroshi Nakashima.)

Pkg `arydshln`

arydshln heavily patches tabular code, so the actual package is not used. **arydshln** is emulated for HTML `tabular`, and reverts to solid rules for SVG math `array` and `tabular` in a `lateximage`.

CSS is not able to display a double-dashed border, so a single-dashed rule is displayed as a single-dashed border, and a double-dashed rule is displayed as a thicker single-dashed border.

for HTML output:

array is required to allow `\newcolumn` below.

```
\RequirePackage{array}

\LWR@ProvidesPackageDrop{arydshln}
```

Ignored, but included for source compatibility:

```
\newdimen\dashlinedash \dashlinedash4pt %
\newdimen\dashlinegap \dashlinegap4pt %
\let\hdashlinewidth\dashlinedash
\let\hdashlinegap\dashlinegap

\def\ADLnullwide{}
\def\ADLsomewide{}
\def\ADLnullwidehline{}
\def\ADLsomewidehline{}

\def\ADLactivate{}
\def\ADLinactivate{}
\newcommand*{\ADLdrawingmode}[1]{}
\newcommand*{\ADLnoshorthanded}{}
\newcommand*{\dashgapcolor}[2][]{}
\newcommand*{\nodashgapcolor}{}
```

In a `lateximage`, revert to solid vertical rules:

```
\appto\LWR@restoreorigformatting{
\newcolumntype{:}{|}%
\newcolumntype{;}[1]{|}%
\LetLtxMacro\hdashline\hline%
```

```
}
```

Some of these macros are already defined as temporary placeholders in the **lwarp** core, so they must be redefined here.

The emulated defaults also work for an emulated print mode inside a `lateximage`:

```
\def\hdashline{
%     \adl@hdashline\adl@ihdashline
    \adl@hdashline\adl@inactivehdl
}
\def\adl@hdashline#1{\noalign{\ifnum0=`}\fi
%         \ifadl@zwhrule \vskip-\arrayrulewidth
%         \else
%             \adl@hline\adl@connect\arrayrulewidth
                \hrule \@height \arrayrulewidth% lwarp
%         \fi
        \@ifnextchar[%
                      {#1}%
                      {#1[%
%                           \dashlinedash/\dashlinegap
                        1pt/1pt
                      ]}}
% \def\adl@ihdashline[#1/#2]{\ifnum0=`{\fi}%
%         \multispan{\adl@columns}\unskip \adl@hcline\z@[#1/#2]%
%         \noalign{\ifnum0=`}\fi
%         \futurelet\@tempa\adl@xhline}
\def\adl@inactivehdl[#1/#2]{
%         \ifadl@zwhrule \vskip-\arrayrulewidth \fi
        \hrule\@height\arrayrulewidth
        \futurelet\@tempa\adl@xhline}
\def\adl@xhline{\ifx\@tempa\hline \adl@ixhline\fi
        \ifx\@tempa\hdashline \adl@ixhline\fi
        \ifnum0=`{\fi}}
\def\adl@ixhline{\vskip\doublerulesep \adl@hline\relax\doublerulesep}
\def\adl@hline#1#2{%
% \@tempcnta#2
%         \global\advance\adl@totalheight\@tempcnta
%         \xdef\adl@rowsL{\adl@rowsL
%                 (#1/\number\@tempcnta);}%
%         \xdef\adl@rowsR{\adl@rowsR
%                 (#1/\number\@tempcnta);}
}

\def\cdashline#1{\noalign{\ifnum0=`}\fi
        \@ifnextchar[%
%                       {\adl@cdline[#1]}%
%                       {\adl@cdline[#1][\dashlinedash/\dashlinegap]}
                      {\adl@inactivecdl[#1]}%
```

```
                        {\adl@inactivecdl[#1][\dashlinedash/\dashlinegap]}
}

\def\adl@inactivecdl[#1-#2][#3]{\ifnum0=‘{\fi}\cline{#1-#2}}
```

File 26 `lwarp-asymptote.sty`

§ 118 Package asymptote

(*Emulates or patches code by* Andy Hammerlindl, John Bowman, Tom Prince.)

Pkg `asymptote` **asymptote** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{asymptote}

\BeforeBeginEnvironment{asy}{\begin{lateximage}}
\AfterEndEnvironment{asy}{\end{lateximage}}

\xpatchcmd{\asyinclude}
    {\begingroup}
    {\begin{lateximage}}
    {}
    {\LWR@patcherror{asymptote}{asyinclude-begingroup}}

\xpatchcmd{\asyinclude}
    {\endgroup}
    {\end{lateximage}}
    {}
    {\LWR@patcherror{asymptote}{asyinclude-endgroup}}
```

File 27 `lwarp-atbegshi.sty`

§ 119 Package atbegshi

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg `atbegshi` Emulated.

for HTML output: Discard all options for **lwarp-atbegshi**:

```
\LWR@ProvidesPackageDrop{atbegshi}[2011/10/05]

\newcommand*{\AtBeginShipout}[1]{}
\newbox\AtBeginShipoutBox
```

```
\newcommand*{\AtBeginShipoutNext}[1]{}
\newcommand*{\AtBeginShipoutFirst}[1]{}
\newcommand*{\AtBeginShipoutDiscard}{}
\newcommand*{\AtBeginShipoutInit}{}
\newcommand*{\AtBeginShipoutAddToBox}[1]{}
\newcommand*{\AtBeginShipoutAddToBoxForeground}[1]{}
\newcommand*{\AtBeginShipoutUpperLeft}[1]{}
\newcommand*{\AtBeginShipoutUpperLeftForeground}[1]{}
\newcommand*{\AtBeginShipoutOriginalShipout}[1]{}
\def\AtBeginShipoutBoxWidth{0pt}
\def\AtBeginShipoutBoxHeight{0pt}
\def\AtBeginShipoutBoxDepth{0pt}

```

File 28 `lwarp-attachfile.sty`

§ 120 Package

attachfile

(*Emulates or patches code by* Scott Pakin.)

Pkg `attachfile` **attachfile** is patched for use by **lwarp**.

⚠ Metadata is ignored for now.

for HTML output:

```
\LWR@ProvidesPackagePass{attachfile}
```

Encloses each icon:

```
\newenvironment*{LWR@attachfile@icon}
{
    \begin{lateximage}*%
        [-attachfile-]%
        [%
            \detokenize\expandafter{\atfi@icon@icon}-%
            \detokenize\expandafter{\atfi@color@rgb}%
        ]%
}
{
    \end{lateximage}
}
```

Each icon is enclosed inside a `LWR@attachfile@icon` environment:

```
\xpretocmd{\atfi@acroGraph}{\LWR@attachfile@icon}{}{}
\xapptocmd{\atfi@acroGraph}{\endLWR@attachfile@icon}{}{}

\xpretocmd{\atfi@acroPaperclip}{\LWR@attachfile@icon}{}{}
```

```
\xapptocmd{\atfi@acroPaperclip}{\endLWR@attachfile@icon}{}{}

\xpretocmd{\atfi@acroPushPin}{\LWR@attachfile@icon}{}{}
\xapptocmd{\atfi@acroPushPin}{\endLWR@attachfile@icon}{}{}

\xpretocmd{\atfi@acroTag}{\LWR@attachfile@icon}{}{}
\xapptocmd{\atfi@acroTag}{\endLWR@attachfile@icon}{}{}
```

Disable PDF file embedding:

```
\DeclareRobustCommand{\atfi@embedfile}[1]{}
```

The displayed output for an `\attachfile` reference:

```
\newcommand*{\LWR@attachfile@appearance}{}

\DeclareRobustCommand{\atfi@set@appearance}[1]{%
    \def\LWR@attachfile@appearance{#1}%
}
```

A file annotation becomes a reference:

```
\DeclareRobustCommand{\atfi@insert@file@annot}[1]{%
    \href{#1}{\LWR@attachfile@appearance}%
}
```

File 29 `lwarp-attachfile2.sty`

§ 121 Package attachfile2

(*Emulates or patches code by* HEIKO OBERDIEK.)

Pkg `attachfile2` **attachfile2** is patched for use by **lwarp**.

⚠ Metadata is ignored for now.

for HTML output:

```
\LWR@ProvidesPackagePass{attachfile2}
```

Adds memory of the selected color:

```
\def\LWR@attachfiletwo@color{}%

\define@key{AtFi}{color}{%
    \def\LWR@attachfiletwo@color{#1}%   lwarp
  \HyColor@AttachfileColor{#1}%
          \atfi@color@tex\atfi@color@inline\atfi@color@annot
```

```
          {attachfile2}{color}%
}
```

Encloses each icon:

```
\newenvironment*{LWR@attachfile@icon}
{
    \begin{lateximage}*%
        [-attachfile-]%
        [%
            \detokenize\expandafter{\atfi@icon@icon}-%
            \detokenize\expandafter{\LWR@attachfiletwo@color}%
        ]%
}
{
    \end{lateximage}
}
```

Each icon is enclosed inside a `LWR@attachfile@icon` environment:

```
\xpretocmd{\atfi@acroGraph}{\LWR@attachfile@icon}{}{}
\xapptocmd{\atfi@acroGraph}{\endLWR@attachfile@icon}{}{}

\xpretocmd{\atfi@acroPaperclip}{\LWR@attachfile@icon}{}{}
\xapptocmd{\atfi@acroPaperclip}{\endLWR@attachfile@icon}{}{}

\xpretocmd{\atfi@acroPushPin}{\LWR@attachfile@icon}{}{}
\xapptocmd{\atfi@acroPushPin}{\endLWR@attachfile@icon}{}{}

\xpretocmd{\atfi@acroTag}{\LWR@attachfile@icon}{}{}
\xapptocmd{\atfi@acroTag}{\endLWR@attachfile@icon}{}{}
```

Disable PDF file embedding:

```
\DeclareRobustCommand{\atfi@embedfile}[1]{}
```

The displayed output for an `\attachfile` reference:

```
\newcommand*{\LWR@attachfile@appearance}{}

\def\atfi@set@appearance@icon{%
    \atfi@set@appearance{\csname atfi@acro\atfi@icon@icon\endcsname}%
}

\DeclareRobustCommand{\atfi@set@appearance}[1]{%
    \def\LWR@attachfile@appearance{#1}%
}
```

A file annotation becomes a reference:

```
\DeclareRobustCommand{\atfi@insert@file@annot}[1]{%
    \href{#1}{\LWR@attachfile@appearance}%
}
```

Modified for text color:

```
\DeclareRobustCommand{\notextattachfile}[2][]{%
  \begingroup
    \atfi@setup{#1}%
    \ifatfi@print
      \leavevmode
      \begingroup
        \HyColor@UseColor\atfi@color@tex
        \LWR@textcurrentcolor{#2}%      lwarp
% \strut
      \endgroup
%     \else
%       \sbox\ltx@zero{#2\strut}%
%       \makebox[\wd0]{}%
    \fi
  \endgroup
}
```

Modified to draw the icon:

```
\DeclareRobustCommand{\noattachfile}[1][]{%
  \begingroup
    \atfi@setup{#1}%
    \atfi@set@appearance@icon
    \ifatfi@print
        \LWR@attachfile@appearance%     lwarp
%       \expandafter
%       \atfi@refxform\csname atfi@appobj@\atfi@icon@icon\endcsname
%     \else
%       \makebox[\atfi@appearancewidth]{}%
    \fi
  \endgroup
}
```

File 30 `lwarp-authblk.sty`

§ 122 Package authblk

(*Emulates or patches code by* Patrick W. Daly.)

Pkg authblk **authblk** is patched for HTML.

package support
⚠ load order **lwarp** supports the native LaTeX titling commands, and also supports the packages **authblk** and **titling**. If both are used, **authblk** should be loaded before **titling**.

\published and \subtitle If using the **titling** package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 62.8.

(*Emulates or patches code by* Patrick W. Daly.)

for HTML output: Require that **authblk** be loaded before **titling**:

```
\@ifpackageloaded{titling}{
\PackageError{lwarp-authblk}
{Package authblk must be loaded before titling}
{Titling appends authblk's author macro, so authblk must be loaded first.}
}
{}
```

Load **authblk**:

```
\LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```
\LetLtxMacro\LWRAB@affil\affil

\renewcommand{\affil}[2][]{%
\LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
}
```

Create an HTML break for an \authorcr:

```
\renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

File 31 `lwarp-axessibility.sty`

§ 123 Package

axessibility

Pkg axessibility **axessibility** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{axessibility}

\newcommand{\wrap}[1]{}
```

File 32 `lwarp-axodraw2.sty`

§ 124 Package axodraw2

(*Emulates or patches code by* John C. Collins, J.A.M. Vermaseren.)

Pkg `axodraw2` **axodraw2** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{axodraw2}

\BeforeBeginEnvironment{axopicture}{\begin{lateximage}[(-axopicture-~\packagediagramname)]}

\AfterEndEnvironment{axopicture}{\end{lateximage}}
```

File 33 `lwarp-backref.sty`

§ 125 Package backref

(*Emulates or patches code by* David Carlisle and Sebastian Rahtz.)

Pkg `backref` **backref** is patched for use by **lwarp**.

⚠ loading Note that **backref** must be explicitly loaded, and is not automatically loaded by **hyperref** when generating HTML output.

for HTML output:

```
\LWR@ProvidesPackagePass{backref}
```

Force the `hyperref` option:

```
\def\backref{}\let\backrefxxx\hyper@section@backref
```

File 34 `lwarp-balance.sty`

§ 126 Package balance

(*Emulates or patches code by* Patrick W. Daly.)

Pkg `balance` Emulated.

for HTML output:

Discard all options for **lwarp-balance**:

```
\LWR@ProvidesPackageDrop{balance}

\newcommand*{\balance}{}
\newcommand*{\nobalance}{}
```

File 35 `lwarp-bibunits.sty`

§ 127 Package

bibunits

(*Emulates or patches code by* Thorsten Hansen.)

Pkg bibunits **bibunits** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{bibunits}

\def\bu@bibdata{\BaseJobname}
```

File 36 `lwarp-bigdelim.sty`

§ 128 Package

bigdelim

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg bigdelim **bigdelim** is used as-is for print or `lateximage`, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the `\multirow` which indicates the actual height of the delimiter. The delimiter character is given a `<span>` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`.

⚠ use `\mrowcell` `\ldelim` and `\rdelim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\ldelim` or `\rdelim`, but not in cells which are above or below the delimiter:

```
\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\{}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

<> a b
left { c d
e f
<> g h

for HTML output: First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case **bigdelim** was not loaded:

```
\let\ldelim\relax
\let\rdelim\relax
```

Next, load the package's new definitions:

```
\LWR@ProvidesPackagePass{bigdelim}
```

`\ldelim` {⟨*1:delimiter*⟩} {⟨*2:#rows*⟩} {⟨*3:width*⟩} [⟨*4:text*⟩]
`\rdelim`

```
\NewDocumentCommand{\LWR@HTML@ldelim}{m m m O{}}{%
\renewcommand{\LWR@multirowborder}{right}%
\multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
}

\LWR@formatted{ldelim}

\NewDocumentCommand{\LWR@HTML@rdelim}{m m m O{}}{%
\renewcommand{\LWR@multirowborder}{left}%
\multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
}

\LWR@formatted{rdelim}
```

File 37 `lwarp-bigstrut.sty`

§ 129 Package bigstrut

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg `bigstrut` **bigstrut** is used as-is for print or `lateximage`, and patched for HTML.

for HTML output:

```
\LWR@ProvidesPackagePass{bigstrut}

\LetLtxMacro\LWR@origbigstrut\bigstrut

\renewcommand\bigstrut[1][x]{}

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\bigstrut\LWR@origbigstrut%
}
```

File 38 `lwarp-blowup.sty`

§ 130 Package blowup

Pkg `blowup` **blowup** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{blowup}

\newcommand*\blowUp[1]{}
```

File 39 `lwarp-booklet.sty`

§ 131 Package booklet

(*Emulates or patches code by* Peter Wilson.)

Pkg `booklet` **booklet** is nullified.

for HTML output:

```
\LWR@ProvidesPackageDrop{booklet}[2009/09/02]

\newdimen\pageseplength
\newdimen\pagesepwidth
\newdimen\pagesepoffset
\newif\ifsidebyside     \sidebysidetrue
\newif\ifuselandscape   \uselandscapefalse
\newif\ifprintoption    \printoptionfalse
\newcommand*{\pagespersignature}[1]{}
\def\magstepminus#1{}
\newcommand*{\target}[3]{}
\newcommand*{\source}[3]{}
\newcommand*{\setpdftargetpages}{}
\newcommand*{\setdvipstargetpages}{}
\newcommand*{\targettopbottom}{}
```

```
\newcommand*{\twoupemptypage}{}
\newcommand*{\twoupclearpage}{}
\newcommand*{\checkforlandscape}{}
```

File 40 `lwarp-bookmark.sty`

§ 132 Package bookmark

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg `bookmark` **bookmark** is emulated.

for HTML output: Discard all options for **lwarp-bookmark**:

```
\LWR@ProvidesPackageDrop{bookmark}

\newcommand*{\bookmarksetup}[1]{}
\newcommand*{\bookmarksetupnext}[1]{}
\newcommand*{\bookmark}[2][]{}
\newcommand*{\bookmarkdefinestyle}[2]{}
\newcommand*{\bookmarkget}[1]{}
\newcommand{\BookmarkAtEnd}[1]{}
```

File 41 `lwarp-booktabs.sty`

§ 133 Package booktabs

(*Emulates or patches code by* Simon Fear.)

Pkg `booktabs` **booktabs** is emulated during HTML output, and used as-is during print output and inside an HTML `lateximage`.

for HTML output: First, forget the placeholder macros:

```
\LetLtxMacro\toprule\relax
\LetLtxMacro\midrule\relax
\LetLtxMacro\cmidrule\cline
\LetLtxMacro\bottomrule\relax
\LetLtxMacro\addlinespace\relax
\LetLtxMacro\morecmidrules\relax
\LetLtxMacro\specialrule\relax

\LWR@ProvidesPackagePass{booktabs}
```

```
\DeclareDocumentCommand{\LWR@HTML@toprule}{o d()}%
    {%
        \IfValueTF{#1}%
            {\LWR@docmidrule[#1]()}{1-\arabic{LWR@tabletotalLaTeXcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[#1]()}{1-\arabic{LWR@tabletotalLaTeXcols}}}%
                {\global\booltrue{LWR@doingtbrule}}%
            }%
    \LWR@getmynexttoken}

\LWR@expandableformatted{toprule}

\DeclareDocumentCommand{\LWR@HTML@midrule}{o d()}%
    {%
        \IfValueTF{#1}%
            {\LWR@docmidrule[#1]()}{1-\arabic{LWR@tabletotalLaTeXcols}}}%
            {%
                \ifbool{FormatWP}%
                {\LWR@docmidrule[#1]()}{1-\arabic{LWR@tabletotalLaTeXcols}}}%
                {\addtocounter{LWR@hlines}{1}}%
            }%
    \LWR@getmynexttoken}

\LWR@expandableformatted{midrule}

\DeclareDocumentCommand{\LWR@HTML@cmidrule}{O{\LWR@cmidrulewidth} d() m}{%
    \LWR@docmidrule[#1](#2){#3}%
    \LWR@getmynexttoken%
}%

\LWR@expandableformatted{cmidrule}

\DeclareDocumentCommand{\LWR@HTML@bottomrule}{o d()}{%
    \IfValueTF{#1}%
        {\LWR@docmidrule[#1]()}{1-\arabic{LWR@tabletotalLaTeXcols}}}%
        {%
            \ifbool{FormatWP}%
            {\LWR@docmidrule[#1]()}{1-\arabic{LWR@tabletotalLaTeXcols}}}%
            {\global\booltrue{LWR@doingtbrule}}%
        }%
    \LWR@getmynexttoken%
}%

\LWR@expandableformatted{bottomrule}

\DeclareDocumentCommand{\LWR@HTML@addlinespace}{o}{}%

\LWR@expandableformatted{addlinespace}

```

```
\DeclareDocumentCommand{\LWR@HTML@morecmidrules}{}{}%

\LWR@expandableformatted{morecmidrules}

\DeclareDocumentCommand{\LWR@HTML@specialrule}{m m m d()}%
    {\LWR@docmidrule[#1](){1-\arabic{LWR@tabletotalLaTeXcols}}\LWR@getmynexttoken}%

\LWR@expandableformatted{specialrule}
```

File 42 `lwarp-bophook.sty`

§ 134 Package bophook

Pkg bophook **bophook** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{bophook}

\newcommand*{\AtBeginPage}[1]{}
\newcommand*{\PageLayout}[1]{}
```

File 43 `lwarp-boxedminipage.sty`

§ 135 Package boxedminipage

Pkg boxedminipage **boxedminipage** is superceded by **boxedminipage2e**.

for HTML output:

```
\LWR@loadnever{boxedminipage}{boxedminipage2e}
```

File 44 `lwarp-boxedminipage2e.sty`

§ 136 Package boxedminipage2e

(*Emulates or patches code by* Scott Pakin.)

Pkg boxedminipage2e **boxedminipage2e** is emulated.

for HTML output: Discard all options for **lwarp-boxedminipage2e**:

```
\LWR@ProvidesPackageDrop{boxedminipage2e}

\newenvironment{boxedminipage}{%
```

```
\begin{BlockClass}{framebox}%
\minipage%
}
{
\endminipage%
\end{BlockClass}
}
```

File 45 `lwarp-breakurl.sty`

§ 137 Package breakurl

(*Emulates or patches code by* Vilar Camara Neto.)

Pkg `breakurl` **breakurl** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{breakurl}

\LetLtxMacro\burl\url

\NewDocumentCommand{\LWR@burlaltb}{O{} +m m}{%
\LWR@ensuredoingapar%
\LWR@subhyperref{#2}%
\LWR@subhyperreftext{#3}%
\endgroup% restore catcodes
}

\newrobustcmd*{\burlalt}{%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@burlaltb%
}

\LetLtxMacro\urlalt\burlalt
```

File 46 `lwarp-breqn.sty`

§ 138 Package breqn

(*Emulates or patches code by* Michael J. Downes, Morten Høgholm.)

Pkg breqn **breqn** is patched for use by **lwarp**.

⚠ darray is not supported.

⚠ MathJax does not support **breqn**.

for HTML output:

```
\LWR@ProvidesPackagePass{breqn}

\setkeys{breqn}{spread={5pt}}

\def\eqnumside{R}
% \def\eqnumplace{T}

\BeforeBeginEnvironment{dmath}{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[-breqn dmath- \mathimagename]
}

\AfterEndEnvironment{dmath}{
    \end{lateximage}\end{BlockClass}
}

\BeforeBeginEnvironment{dmath*}{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[-breqn dmath*- \mathimagename]
}

\AfterEndEnvironment{dmath*}{
    \end{lateximage}\end{BlockClass}
}

\BeforeBeginEnvironment{dseries}{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[-breqn dseries- \mathimagename]
}

\AfterEndEnvironment{dseries}{
    \end{lateximage}\end{BlockClass}
}

\BeforeBeginEnvironment{dseries*}{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
```

```
    \begin{lateximage}[-breqn dseries*- \mathimagename]
}

\AfterEndEnvironment{dseries*}{
    \end{lateximage}\end{BlockClass}
}

\BeforeBeginEnvironment{dgroup}{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[-breqn dgroup- \mathimagename]
}

\AfterEndEnvironment{dgroup}{
    \end{lateximage}\end{BlockClass}
}

\BeforeBeginEnvironment{dgroup*}{
    \begin{BlockClass}{displaymath}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[-breqn dgroup*- \mathimagename]
}

\AfterEndEnvironment{dgroup*}{
    \end{lateximage}\end{BlockClass}
}
```

File 47 `lwarp-bxpapersize.sty`

§ 139 Package bxpapersize

Pkg `bxpapersize` **bxpapersize** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{bxpapersize}

\providecommand*\papersizesetup{\bxpapersizesetup}
\newcommand*\bxpapersizesetup[1]{}
```

File 48 `lwarp-bytefield.sty`

§ 140 Package bytefield

(*Emulates or patches code by* Scott Pakin.)

Pkg bytefield **bytefield** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{bytefield}

\BeforeBeginEnvironment{bytefield}{\begin{lateximage}[(-bytefield--\packagediagramname)]}

\AfterEndEnvironment{bytefield}{\end{lateximage}}
```

File 49 `lwarp-cancel.sty`

§ 141 Package cancel

Pkg cancel **cancel** is used as-is for SVG math, and emulated for HTML text output.

for HTML output:

```
\LWR@origRequirePackage{lwarp-xcolor}% for \convertcolorspec
\LWR@ProvidesPackagePass{cancel}
```

`\cancelto` is math-only, so is used as-is.

```
\LetLtxMacro\LWR@origcancel\cancel
\LetLtxMacro\LWR@origbcancel\bcancel
\LetLtxMacro\LWR@origxcancel\xcancel

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\cancel\LWR@origcancel%
\LetLtxMacro\bcancel\LWR@origbcancel%
\LetLtxMacro\xcancel\LWR@origxcancel%
}
```

`\LWR@cancelcolor` {⟨*text*⟩} {⟨*color*⟩} {⟨*class*⟩} {⟨*colorstyle*⟩} {⟨*FormatWPstyle*⟩}

Add colors if not empty:

```
\newcommand{\LWR@cancelcolor}[5]{%
\ifcsempty{#2}%
{\LWR@HTMLtextstyle{#5}{#3}{#1}}%
```

```
{\LWR@htmlspanclass[#5;#4:\LWR@origpound\LWR@tempcolor]{#3}{#1}}%
}
```

\cancel {⟨*text*⟩}

```
\DeclareRobustCommand{\cancel}[1]{%
\begingroup%
\CancelColor%
\LWR@findcurrenttextcolor%
\color{black}%
\LWR@cancelcolor{#1}{LWR@tempcolor}{sout}{text-decoration-color}%
    {text-decoration:line-through}%
\endgroup%
}

\LetLtxMacro\bcancel\cancel
\LetLtxMacro\xcancel\cancel
```

File 50 `lwarp-canoniclayout.sty`

§ 142 Package canoniclayout

Pkg canoniclayout **canoniclayout** is ignored.

for HTML output:

S
```
\LWR@ProvidesPackageDrop{canoniclayout}

\newcommand*{\currentfontletters}{}
\newcommand*{\charactersperpage}{}
```

File 51 `lwarp-caption.sty`

§ 143 Package caption

(*Emulates or patches code by* Axel Sommerfeldt.)

Pkg caption **caption** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{caption}

\renewcommand\caption@ibox[3]{%
  \@testopt{\caption@iibox{#1}{#2}{#3}}{%
%          \wd\@tempboxa%
         \linewidth% lwarp
  }%
```

```
% \LWR@traceinfo{caption@ibox: done}%
}

\long\def\caption@iibox#1#2#3[#4]{%
  \@testopt{\caption@iiibox{#1}{#2}{#3}{#4}}\captionbox@hj@default
}

\long\def\caption@iiibox#1#2#3#4[#5]#6{%
%   \setbox\@tempboxa\hbox{#6}%
  \begingroup
  #1*% set \caption@position
  \caption@iftop{%
    \LWR@traceinfo{caption@iiibox top}%
    \endgroup
    \parbox[t]{#4}{%
      #1\relax
      \caption@setposition t%
%       \vbox{\caption#2{#3}}%
      {\caption#2{#3}}% lwarp
%       \captionbox@hrule
%       \csname caption@hj@#5\endcsname
%       \unhbox\@tempboxa
       #6% lwarp
    }%
  }{%
    \LWR@traceinfo{caption@iiibox bottom}%
    \endgroup
    \parbox[b]{#4}{%
      #1\relax
      \caption@setposition b%
%       \csname caption@hj@#5\endcsname
%       \unhbox\@tempboxa
        #6% lwarp
%       \captionbox@hrule
%       \vtop{\caption#2{#3}}}%
      {\caption#2{#3}}% lwarp
     }%
  }%
\LWR@traceinfo{caption@iiibox: done}%
}

\def\caption@caption{%
  \caption@iftype
    {%
     \caption@checkgrouplevel\@empty\caption
     \caption@star
       {\caption@refstepcounter\@captype}%
       {\caption@dblarg{\@caption\@captype}}}%
    {\caption@Error{\noexpand\caption outside float}%
```

```
     \caption@gobble}%
}

\long\def\caption@@caption#1[#2]#3{%

  \ifcaption@star \else
    \caption@prepareanchor{#1}{#2}%
    \memcaptioninfo{#1}{\csname the#1\endcsname}{#2}{#3}%
    \@nameuse{nag@hascaptiontrue}%
  \fi

  \par
  \caption@beginex{#1}{#2}{#3}%
    \caption@setfloatcapt{%
      \caption@boxrestore
      \if@minipage
        \@setminipage
      \fi
      \caption@normalsize
      \ifcaption@star
        \let\caption@makeanchor\@firstofone
      \fi
      \@makecaption{\csname fnum@#1\endcsname}%
                   {\ignorespaces\caption@makeanchor{#3}}\par
      \caption@if@minipage\@minipagetrue\@minipagefalse}%
  \caption@end%
}
```

`\caption@@@make` {⟨*caption label*⟩} {⟨*caption text*⟩}

```
\renewcommand\caption@@@make[2]{%
\LWR@startpars% lwarp
%   \sbox\@tempboxa{#1}%
%   \ifdim\wd\@tempboxa=\z@
%     \let\caption@lsep\relax
%   \fi
  \caption@ifempty{#2}{%
    \let\caption@lsep\@empty
    \let\caption@tfmt\@firstofone
  }%
  \@setpar{\LWR@closeparagraph\@@par}% lwarp
  \caption@applyfont
  \caption@fmt
    {\ifcaption@star\else
       \begingroup
         \captionlabelfont
         #1%
       \endgroup
     \fi}%
```

```
    {\ifcaption@star\else
       \begingroup
         \caption@iflf\captionlabelfont
         \relax\caption@lsep
       \endgroup
     \fi}%
    {{\captiontextfont
      \caption@ifstrut
        {\vrule\@height\ht\strutbox\@width\z@}%
        {}%
      \nobreak\hskip\z@skip % enable hyphenation
      \caption@tfmt{#2}
    \LWR@ensuredoingapar% lwarp
      \caption@ifstrut
        {\ifhmode\@finalstrut\strutbox\fi}%
        {}%
      \par}}
\LWR@stoppars% lwarp
}
```

`\caption@@make@` {⟨⟩} {⟨⟩}

```
\renewcommand{\caption@@make@}[2]{%
  \caption@stepthecounter
  \caption@beginhook
      \caption@@@make{#1}{#2}%
  \caption@endhook
}

% \DeclareCaptionBox{none}{#2}
\DeclareCaptionBox{parbox}{%
#2%
}
\DeclareCaptionBox{colorbox}{%
#2%
}
```

File 52 `lwarp-caption2.sty`

§ 144 Package caption2

Pkg `caption2` **caption2** is not used. The user is recommended to use **caption** instead.

for HTML output:

```
\LWR@loadnever{caption2}{caption}
```

File 53 `lwarp-cases.sty`

§ 145 Package cases

(*Emulates or patches code by* Donald Arseneau.)

Pkg `cases` **cases** is patched for use by **lwarp**.

⚠ MathJax does not support **cases**.

for HTML output:

```
\LWR@ProvidesPackagePass{cases}

\BeforeBeginEnvironment{numcases}{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[-cases- \mathimagename]
}

\AfterEndEnvironment{numcases}{
    \end{lateximage}\end{BlockClass}
}

\BeforeBeginEnvironment{subnumcases}{
    \begin{BlockClass}{displaymathnumbered}
    \LWR@newautoidanchor%
    \booltrue{LWR@indisplaymathimage}%
    \begin{lateximage}[-cases- \mathimagename]
}

\AfterEndEnvironment{subnumcases}{
    \end{lateximage}\end{BlockClass}
}
```

File 54 `lwarp-ccaption.sty`

§ 146 Package ccaption

Pkg `ccaption` **ccaption** is not used. The user is recommended to use **caption** instead.

for HTML output:

```
\LWR@loadnever{ccaption}{caption}
```

File 55 `lwarp-changebar.sty`

§ 147 Package changebar

Pkg changebar **changebar** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{changebar}

\newcommand*{\cbstart}{}
\newcommand*{\cbend}{}
\newenvironment*{\changebar}{}{}
\newcommand*{\cbdelete}{}
\newcommand*{\nochnagebars}{}
\newcommand*{\cbcolor}[1]{}
\newlength{\changebarwidth}
\newlength{\deletebarwidth}
\newlength{\changebarsep}
\newcounter{changebargrey}
```

File 56 `lwarp-changepage.sty`

§ 148 Package changepage

(*Emulates or patches code by* Peter Wilson.)

Pkg changepage **changepage** is emulated.

for HTML output: Discard all options for **lwarp-changepage**:

```
\LWR@ProvidesPackageDrop{changepage}

\newif\ifoddpage
\DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
\DeclareRobustCommand{\changetext}[5]{}
\DeclareRobustCommand{\changepage}[9]{}

\@ifundefined{adjustwidth}{
\newenvironment{adjustwidth}[2]{}{}
\newenvironment{adjustwidth*}[2]{}{}
}{
\renewenvironment{adjustwidth}[2]{}{}
\renewenvironment{adjustwidth*}[2]{}{}
}
```

```
\DeclareDocumentCommand{\strictpagecheck}{}{}
\DeclareDocumentCommand{\easypagecheck}{}{}
```

File 57 `lwarp-chngpage.sty`

§ 149 Package chngpage

(*Emulates or patches code by* Peter Wilson.)

Pkg chngpage **chngpage** is emulated.

for HTML output: Discard all options for **lwarp-chngpage**:

```
\LWR@ProvidesPackageDrop{chngpage}
\LWR@origRequirePackage{changepage}
```

File 58 `lwarp-chappg.sty`

§ 150 Package chappg

(*Emulates or patches code by* Robin Fairbairns.)

Pkg chappg **chappg** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{chappg}
\renewcommand{\pagenumbering}[2][]{}
\providecommand{\chappgsep}{--}
```

File 59 `lwarp-chapterbib.sty`

§ 151 Package chapterbib

(*Emulates or patches code by* Donald Arseneau.)

Pkg chapterbib **chapterbib** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{chapterbib}
\xdef\@savedjobname{\BaseJobname}
\let\@currentipfile\@savedjobname
```

File 60 `lwarp-chemfig.sty`

§ 152 Package

chemfig

(*Emulates or patches code by* Christian Tellechea.)

Pkg `chemfig` **chemfig** is patched for use by **lwarp**.

The images are not hashed because they depend on external settings which may be changed at any time, and are unlikely to be reused inline anyhow.

for HTML output:

```
\LWR@ProvidesPackagePass{chemfig}

\LetLtxMacro\LWR@chemfig@origchemfig\chemfig

\DeclareDocumentCommand\chemfig{s O{} O{} m}{%
    \begin{lateximage}[(-chemfig-~\packagediagramname)]%
    \IfBooleanTF{#1}{%
        \LWR@chemfig@origchemfig*[#2][#3]{#4}%
    }{%
        \LWR@chemfig@origchemfig[#2][#3]{#4}%
    }
    \end{lateximage}%
}

\LetLtxMacro\LWR@chemfig@origCF@lewis@b\CF@lewis@b

\def\CF@lewis@b#1#2{%
\begin{lateximage}[(-chemfig-~\packagediagramname)]%
\LWR@chemfig@origCF@lewis@b{#1}{#2}%
\end{lateximage}%
}

\preto{\schemestart}{\begin{lateximage}[(-chemfig-~\packagediagramname)]}
\appto{\CF@schemestop}{\end{lateximage}}

\LetLtxMacro\LWR@chemfig@origchemleft\chemleft

\def\chemleft#1#2\chemright#3{%
\begin{lateximage}[(-chemfig-~\packagediagramname)]%
\LWR@chemfig@origchemleft#1#2\chemright#3%
\end{lateximage}%
}

\LetLtxMacro\LWR@chemfig@origchemup\chemup
```

```
\def\chemup#1#2\chemdown#3{%
\begin{lateximage}[(-chemfig-~\packagediagramname)]%
\LWR@chemfig@origchemup#1#2\chemdown#3%
\end{lateximage}%
}
```

File 61 `lwarp-chemformula.sty`

§ 153 Package

chemformula

(*Emulates or patches code by* Clemens Niederberger.)

Pkg `chemformula`

chemformula is patched for use by **lwarp**.

The SVG images are hashed according to contents and local options. Global options are assumed to be constant document-wide.

⚠ **chemformula** with MathJax

chemformula works best without MathJax. If MathJax is used, `\displaymathother` must be used before `array`, and then `\displaymathnormal` may be used after. (The **chemformula** package adapts to `array`, but does not know about MathJax, and MathJax does not know about **chemformula**.)

While using MathJax, `\displaymathother` may also be used for other forms of display and inline math which contain **chemformula** expressions.

for HTML output:

```
\LWR@ProvidesPackagePass{chemformula}[2017/03/23]
```

```
\ExplSyntaxOn
```

`\ch` Enclose in an inline SVG image or MathJax. The `alt` tag is is the contents of the `\ch` expression. The filename is hashed, and also has additional hashing information based on the local options.

```
\RenewDocumentCommand \ch { O{}m }
 {%
```

To work inside `align` with `\displaymathother`, a simple version must be used to work with **chemformula**'s adaptation to `align`.

```
    \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%  lwarp
    {
        \chemformula_ch:nn {#1} {#2}%   original
    }
```

If used as the outer level, must temporarily ensure MathJax is disabled:

```
    {
        \begingroup%
        \boolfalse{mathjax}%
```

An inline image is used, adjusted for the baseline:

```
        \LWR@subsingledollar*{% lwarp
            \textbackslash{}ch\{\LWR@HTMLsanitize{#2}\}% alt text
        }{%
           \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}% add'l hashing
        }%
        {%
            \chemformula_ch:nn {#1} {#2}%   original
        }%
        \endgroup%
    }
}
```

\chcpd Similar to \ch.

```
\cs_gset_protected:Npn \chemformula_chcpd:nn #1#2
  {
    \begingroup%
    \boolfalse{mathjax}%
    \LWR@subsingledollar*{% lwarp
        \textbackslash{}chcpd\{\LWR@HTMLsanitize{#2}\}%
    }{%
        \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
    }{% original
    \group_begin:
      \tl_if_blank:nF {#2}
        {
          \keys_set:nn {chemformula} {#1}
          \__chemformula_save_catcodes:
          \__chemformula_sanitize:Nn
            \l__chemformula_chemformula_tmpa_tl
            {#2}
          \__chemformula_input_compound_no_check:NV
            \l__chemformula_compound_tl
            \l__chemformula_chemformula_tmpa_tl
          \__chemformula_prepare_output:N \l__chemformula_compound_tl
          \chemformula_write:V \l__chemformula_compound_tl
        }
    \group_end:
    }
    \endgroup
  }
```

\charrow If standalone, appears in a regular lateximage.

```
\RenewDocumentCommand \charrow { mO{}O{} }
{
    \begin{lateximage}[(-chemformula- charrow)]
    \group_begin:
      \__chemformula_draw_arrow:nnn {#1} {#2} {#3}
    \group_end:
    \end{lateximage}
}
```

\chname If standalone, appears in a regular lateximage, hashed according to contents.

```
\RenewDocumentCommand \chname { R(){}R(){} }
 {
   \begin{lateximage}*[%
       \textbackslash{}chname(\LWR@HTMLsanitize{#1})(\LWR@HTMLsanitize{#2})
   ]%
       \chemformula_chwritebelow:nn {#1} {#2}
   \end{lateximage}
 }
```

\chlewis Placed inline, hashed according to contents and options.

```
\RenewDocumentCommand \chlewis { O{}mm }
 {
   \begingroup%
   \boolfalse{mathjax}%
   \LWR@subsingledollar*{\textbackslash{}chlewis\{#2\}\{#3\}}%
   {
       \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
   }{
       \chemformula_lewis:nnn {#1} {#2} {#3}
   }
   \endgroup%
 }
```

lwarp redefines the $ character, so special handling is required to escape math expressions inside \ch.

This boolean tracks a new kind of escaped math:

```
\bool_new:N      \l__chemformula_first_last_LWRdollar_bool
```

\chemformula_input_escape_math

Adds additional escaping for the new dollar definition:

```
\cs_gset_protected:Npn \__chemformula_input_escape_math:n #1
 {
```

```
    \__chemformula_first_last_math:n {#1}
    \bool_if:NT \l__chemformula_first_last_dollar_bool
      {
        \bool_set_true:N \l__chemformula_first_last_math_bool
        \__chemformula_read_escape_dollar:w #1 \q_nil
      }
    \bool_if:NT \l__chemformula_first_last_mathbraces_bool
      {
        \bool_set_true:N \l__chemformula_first_last_math_bool
        \__chemformula_read_escape_mathbraces:w #1 \q_nil
      }
```

Added by **lwarp**:

```
    \bool_if:NT \l__chemformula_first_last_LWRdollar_bool%      lwarp
      {
        \bool_set_true:N \l__chemformula_first_last_math_bool%  lwarp
        \__chemformula_read_escape_LWRdollar:w #1 \q_nil%       lwarp
      }
  }
```

`\chemformula_read_escape_LWRdollar`

The following parses the contents inside the new dollars.

lwarp keeps the dollar as its original math shift until the document starts. While **chemmacros** is being patched, the dollar must temporarily be set to its new meaning during the following definition.

```
\begingroup
\catcode`\$=\active

\cs_new_protected:Npn \__chemformula_read_escape_LWRdollar:w $#1$ \q_nil
{
    \__chemformula_read_escape_math:n {#1}
}

\endgroup
```

`\chemformula_bool_set_if_first_last`

The following looks at the first and last tokens for delimiters to escape math inside `\ch`. The original definition is modified to look for the control sequences which are used by the new meaning of `$`.

```
\cs_new_protected:Npn \__chemformula_bool_cs_set_if_first_last:NnNN #1#2#3#4
  {
    \int_zero:N \l__chemformula_tmpa_int
    \int_zero:N \l__chemformula_tmpb_int
```

```
    \int_set:Nn \l__chemformula_tmpa_int { \tl_count:n {#2} }
    \tl_map_inline:nn {#2}
      {
        \int_incr:N \l__chemformula_tmpb_int
        \int_compare:nT { \l__chemformula_tmpb_int = 1 }
          {
```

At the start, the `cs_` version compares control sequences:

```
            \ifdefstrequal{##1}{#3}% lwarp
                {
                    \bool_set_true:N #1
                }% lwarp
                {}
          }
```

At the end, compare more control sequences:

```
        \int_compare:nT { \l__chemformula_tmpb_int = \l__chemformula_tmpa_int }
          {
            \ifdefstrequal{##1}{#4}
                {}
                {
                    \bool_set_false:N #1
                }
          }
      }
}
```

`\chemformula_first_last_math`

Modified to check for the new meaning of $ at first/last:

```
\cs_gset_protected:Npn \__chemformula_first_last_math:n #1
 {
   \bool_set_false:N \l__chemformula_first_last_math_bool
   \bool_set_false:N \l__chemformula_first_last_dollar_bool
   \bool_set_false:N \l__chemformula_first_last_LWRdollar_bool%    lwarp
   \bool_set_false:N \l__chemformula_first_last_mathbraces_bool
   \__chemformula_bool_set_if_first_last:Nnnn
     \l__chemformula_first_last_dollar_bool
     {#1}
     { $ } { $ }
   \bool_if:NF \l__chemformula_first_last_dollar_bool
     {
       \__chemformula_bool_set_if_first_last:Nnnn
         \l__chemformula_first_last_mathbraces_bool
         {#1}
         { \( } { \) }
```

Added by **lwarp**:

```
149          \bool_if:NF \l__chemformula_first_last_mathbraces_bool%   lwarp
150            {
151                \__chemformula_bool_cs_set_if_first_last:NnNN
152                \l__chemformula_first_last_LWRdollar_bool
153                {#1}
154                { \LWR@newsingledollar } { \LWR@newsingledollar }
155            }%  lwarp
156      }
157  }

158 \ExplSyntaxOff
```

File 62 `lwarp-chemgreek.sty`

§ 154 Package chemgreek

(*Emulates or patches code by* Clemens Niederberger.)

Pkg `chemgreek` **chemgreek** is patched for use by **lwarp**.

Greek symbols
⚠ package selection

To use text-mode symbols, use packages **textalpha** or **textgreek**. Using the other packages supported by **chemgreek** will result in math-mode greek characters, which will result in SVG images being used. These images will be hashed.

⚠ XeLaTeX, LuaLaTeX

If using XeLaTeX or LuaLaTeX, select the `fontspec` mapping:

```
\selectchemgreekmapping{fontspec}
```

for HTML output:

```
1 \LWR@ProvidesPackagePass{chemgreek}[2016/02/10]

2 \ExplSyntaxOn
3
4 \cs_gset_protected:Npn \chemgreek_text:n #1
5   { { \text {#1} } }
6
7 \appto\LWR@restoreorigformatting{%
8 \cs_set_protected:Npn \chemgreek_text:n #1%
9   { \ensuremath { \text {#1} } }%
10 }
11
12 \ExplSyntaxOff
```

File 63 `lwarp-chemmacros.sty`

§ 155 Package

chemmacros

(*Emulates or patches code by* Clemens Niederberger.)

Pkg `chemmacros` **chemmacros** is patched for use by **lwarp**.

for HTML output:

```
1 \LWR@ProvidesPackagePass{chemmacros}
```

SVG file hashing assumes that the relevent options are constant for the entire document.

§ 155.1 Changes to the user's document

⚠ `\makepolymerdelims` When using `\makepolymerdelims`, enclose the entire expression inside a `polymerdelims` environment, such as (from the **chemmacros** manual):

```
\begin{polymerdelims}
\chemfig{-[@{op,.75}]CH_2-CH(-[6]Cl)-[@{cl,0.25}]}
\makepolymerdelims{5pt}[27pt]{op}{cl}
\end{polymerdelims}
```

⚠ redox reactions Redox reactions must be enclosed inside a `redoxreaction` environment. For print output, extra space must be included above and/or below the result, so they are declared as arguments to the environment, instead of being manually entered as per the **chemmacros** manual. For HTML output, the extra space is ignored and a `lateximage` is used instead.

```
\begin{redoxreaction}{7mm}{7mm}
\OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b){oxidation}
\end{redoxreaction}
```

§ 155.2 Code

§ 155.3 Loading modules

Patching **chemmacros** modules must be done `\AtBeginDocument`, since modules are invoked by the user in the preamble, and each patch is only done if the module is loaded.

```
\ExplSyntaxOn

\newcommand{\@ifchemmacrosmoduleloaded}[1]{%
\@ifl@aded{\c__chemmacros_module_extension_tl}{\c__chemmacros_module_prefix_tl.#1}%
}

\ExplSyntaxOff
```

§ 155.4 New environments

`\makepolymerdelims` and redox reactions must be enclosed in a `lateximage` during HTML output. These environments are provided here in HTML mode, and in the **lwarp** core in print mode, as a high-level semantic syntax which automatically embeds the contents in a `lateximage` with an appropriate `alt` tag.

Env `polymerdelims`

```
\DeclareDocumentEnvironment{polymerdelims}{}
{\begin{lateximage}[(-chemmacros- polymer)]}
{\end{lateximage}}
```

Env `redoxreaction` `{`⟨*space above*⟩`}` `{`⟨*space below*⟩`}`

For HTML output, the above and below space is ignored, and a `lateximage` is used instead. For the print output version, see section 82.

```
\DeclareDocumentEnvironment{redoxreaction}{m m}
{\begin{lateximage}[(-chemmacros- redoxreaction)]}
{\end{lateximage}}
```

```
\ExplSyntaxOn
```

§ 155.5 Acid-base

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{acid-base}{
\PackageInfo{lwarp}{Patching~chemmacros~module~acid-base}

\cs_gset_protected:Npn \chemmacros_p:n #1
  {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{
        \textbackslash{}p\{\LWR@HTMLsanitize{#1}\}
    }{
        chemmacrosp\protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
    }{
    \group_begin:
      \mbox
        {
          \chemmacros_p_style:n {p}
          \ensuremath {#1}
        }
    \group_end:
    }
    \endgroup
  }

\RenewDocumentCommand \pH  {} {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}pH}{chemmacros}{
        \chemmacros_p:n { \chemmacros_chemformula:n {H} }
    }
    \endgroup
}

\RenewDocumentCommand \pOH {} {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}pOH}{chemmacros}{
        \chemmacros_p:n { \chemmacros_chemformula:n {OH} }
    }
    \endgroup
}

\RenewDocumentCommand \pKa {O{}}
  {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}pKa{[}#1{]}}{chemmacros #1}{
        \chemmacros_p:n
```

```
        {
            \Ka \ifblank {#1} {}
            { {} \c_math_subscript_token { \chemmacros_bold:n {#1} } }
        }
    }
    \endgroup
  }

\RenewDocumentCommand \pKb {O{}}
  {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}pKb{[}#1{]}}{chemmacros #1}{
        \chemmacros_p:n
        {
            \Kb \ifblank {#1} {}
            { {} \c_math_subscript_token { \chemmacros_bold:n {#1} } }
        }
    }
    \endgroup
  }

\LetLtxMacro\LWR@chemmacros@origKa\Ka
\renewcommand*{\Ka}{%
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}Ka}{chemmacros}{%
        \LWR@chemmacros@origKa%
    }%
    \endgroup
}

\LetLtxMacro\LWR@chemmacros@origKb\Kb
\renewcommand*{\Kb}{%
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}Kb}{chemmacros}{%
        \LWR@chemmacros@origKb%
    }%
    \endgroup
}

\LetLtxMacro\LWR@chemmacros@origKw\Kw
\renewcommand*{\Kw}{%
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}Kw}{chemmacros}{
        \LWR@chemmacros@origKw
    }
    \endgroup
```

```
}

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.6 Charges

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{charges}{
\PackageInfo{lwarp}{Patching~chemmacros~module~charges}

\cs_gset_protected:Npn \fplus  {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}fplus}{chemmacros}
    { \LWR@origensuredmath{\chemformula_fplus:}  }
    \endgroup
}
\cs_gset_protected:Npn \fminus {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}fminus}{chemmacros}
    { \LWR@origensuredmath{\chemformula_fminus:} }
    \endgroup
}

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.7 Nomenclature

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{nomenclature}{
\PackageInfo{lwarp}{Patching~chemmacros~module~nomenclature}

\cs_gset_protected:Npn \chemmacros_charge:n #1
  {
    \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}
    {\chemmacros_chemformula:n { {}^{#1} }}
    {
        \ifmmode
            {\chemmacros_chemformula:n { {}^{#1} }}
        \else
            { \textsuperscript{\ensuremath{#1}} }
        \fi
    }
  }


```

```
\LetLtxMacro\LWR@chemmacros@origchemprime\chemprime

\protected\def\chemprime { \HTMLunicode{2032} }

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\chemprime\LWR@chemmacros@origchemprime%
}

\ChemCompatibilityFrom{5.8}
\cs_gset_protected:Npn \__chemmacros_cip:n #1
  {
    \tl_set:Nn \l__chemmacros_tmpa_tl {#1}
    \int_step_inline:nnnn {0} {1} {9}
      {
        \tl_replace_all:Nnn \l__chemmacros_tmpa_tl
          {##1}
          { { \l__chemmacros_cip_number_tl ##1} }
      }
    {
        \l__chemmacros_cip_inner_tl
        \LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
            \l__chemmacros_tmpa_tl
        }}% lwarp
    }
  }
\EndChemCompatibility

\RenewDocumentCommand \Sconf { O{S} } {
\begin{lateximage}[\textbackslash{}Sconf{[}#1{]}]
    \chemmacros_sconf:n {#1}
\end{lateximage}
}

\RenewDocumentCommand \Rconf { O{R} } {
\begin{lateximage}[\textbackslash{}Rconf{[}#1{]}]
    \chemmacros_rconf:n {#1}
\end{lateximage}
}

\cs_gset_protected:Npn \chemmacros_hapto:n #1
  {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}hapto\{#1\}}{chemmacros}{
        \chemmacros_coordination_symbol:nnnn
        { \l__chemmacros_coord_use_hyphen_bool }
        {
            \chemmacros_if_compatibility:nnTF {>} {5.7}
            { \c_true_bool }
            { \c_false_bool }
```

```
        }
        { \chemeta }
        {#1}
    }
    \endgroup
  }

\cs_gset_protected:Npn \chemmacros_dento:n #1
  {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}dento\{#1\}}{chemmacros}{
        \chemmacros_coordination_symbol:nnnn
        { \l__chemmacros_coord_use_hyphen_bool }
        {
            \chemmacros_if_compatibility:nnTF {>} {5.7}
            { \c_true_bool }
            { \c_false_bool }
        }
        { \chemkappa }
        {#1}
    }
    \endgroup
  }

\cs_gset_protected:Npn \chemmacros_bridge:n #1
  {
    \begingroup
    \boolfalse{mathjax}
    \LWR@subsingledollar*{\textbackslash{}bridge\{#1\}}{chemmacros}{
        \chemmacros_coordination_symbol:nnnn
        { \l__chemmacros_coord_use_hyphen_bool }
        { \l__chemmacros_bridge_super_bool }
        { \chemmu }
        {#1}
    }
    \endgroup
  }
}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.8 Particles

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{particles}{
\PackageInfo{lwarp}{Patching~chemmacros~module~particles}

\cs_gset_protected:Npn \chemmacros_declare_nucleophile:Nn #1#2
  {
```

```
    \cs_set_protected:cpn {__chemmacros_ \chemmacros_remove_backslash:N #1:}
      {
        \bool_if:NTF \l__chemmacros_nucleophile_elpair_bool
          {
            \chemmacros_elpair:n { #2 }
            \chemmacros_if_compatibility:nnT {>=} {5.3}
              { \skip_horizontal:N \l__chemmacros_nucleophile_dim }
            \chemmacros_chemformula:n { {}^{-} }
          }
          { \chemmacros_chemformula:n { #2^{-} } }
      }
    \DeclareDocumentCommand #1 {o}
      {%
        \begin{lateximage}%
        \group_begin:%
          \IfNoValueF {##1}%
            { \chemmacros_set_keys:nn {particles} {##1} }%
          \use:c {__chemmacros_ \chemmacros_remove_backslash:N #1:}%
        \group_end:%
        \end{lateximage}%
      }
  }

\RenewChemNucleophile \Nuc {Nu}
\RenewChemNucleophile \ba  {ba}

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.9 Phases

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{phases}{
\PackageInfo{lwarp}{Patching~chemmacros~module~phases}

\cs_undefine:N \chemmacros_phase:n
\cs_new_protected:Npn \chemmacros_phase:n #1
  {
    \chemmacros_leave_vmode:
    \bool_if:NTF \l__chemmacros_phases_sub_bool
      {
        \ifnumequal{\value{LWR@lateximagedepth}}{0}
        {
            \textsubscript{ (#1) }
        }
        {
            \chemformula_subscript:n { (#1) }
        }
      }
```

```
      {
        \skip_horizontal:N \l__chemmacros_phases_space_dim
        \chemmacros_text:n { (#1) }
      }
  }

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.10 Mechanisms

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{mechanisms}{
\PackageInfo{lwarp}{Patching~chemmacros~module~mechanisms}

\chemmacros_define_keys:nn {textmechanisms}
  {
    type      .choice: ,
    type /    .code:n    =
      {
        \__chemmacros_set_mechanisms:nnn { S }
          {
              \textsubscript{N}
          }
          { }
      } ,
    type / 1  .code:n    =
      {
        \__chemmacros_set_mechanisms:nnn { S }
          {
              \textsubscript{N}
            1
          }
          { }
      } ,
    type / 2  .code:n    =
      {
        \__chemmacros_set_mechanisms:nnn { S }
          {
              \textsubscript{N}
            2
          }
          { }
      } ,
    type / se .code:n    =
      {
        \__chemmacros_set_mechanisms:nnn { S }
          {
              \textsubscript{E}
```

```
342           }
343           { }
344       } ,
345     type / 1e .code:n    =
346       {
347         \__chemmacros_set_mechanisms:nnn { S }
348           {
349               \textsubscript{E}
350             1
351           }
352           { }
353       } ,
354     type / 2e .code:n    =
355       {
356         \__chemmacros_set_mechanisms:nnn { S }
357           {
358               \textsubscript{E}
359             2
360           }
361           { }
362       } ,
363     type / ar .code:n    =
364       {
365         \__chemmacros_set_mechanisms:nnn { S }
366           {
367               \textsubscript{E}
368           }
369           { Ar - }
370       } ,
371     type / e  .code:n    =
372       { \__chemmacros_set_mechanisms:nnn { E } { } { } } ,
373     type / e1 .code:n    =
374       { \__chemmacros_set_mechanisms:nnn { E } { 1 } { } } ,
375     type / e2 .code:n    =
376       { \__chemmacros_set_mechanisms:nnn { E } { 2 } { } } ,
377     type / cb .code:n    =
378       {
379         \__chemmacros_set_mechanisms:nnn { E }
380           {
381             1
382               \textsubscript{cb}
383           }
384           { }
385       } ,
386     type      .default:n =
387   }
388
389 \cs_gset_protected:Npn \chemmacros_mechanisms:n #1
390   {
391     \tl_if_blank:nTF {#1}
```

```
      { \chemmacros_set_keys:nn {textmechanisms} { type } }
      { \chemmacros_set_keys:nn {textmechanisms} { type = #1 } }
    \mbox
      {
        \tl_use:N \l__chemmacros_mechanisms_ar_tl
        \tl_use:N \l__chemmacros_mechanisms_type_tl
        \tl_use:N \l__chemmacros_mechanisms_mol_tl
      }
  }

\appto\LWR@restoreorigformatting{%
\cs_set_protected:Npn \chemmacros_mechanisms:n #1%
  {%
    \tl_if_blank:nTF {#1}%
      { \chemmacros_set_keys:nn {mechanisms} { type } }%
      { \chemmacros_set_keys:nn {mechanisms} { type = #1 } }%
    \mbox%
      {%
        \tl_use:N \l__chemmacros_mechanisms_ar_tl%
        \tl_use:N \l__chemmacros_mechanisms_type_tl%
        \tl_use:N \l__chemmacros_mechanisms_mol_tl%
      }%
  }%
}

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.11 Newman

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{newman}{
\PackageInfo{lwarp}{Patching~chemmacros~module~newman}

\RenewDocumentCommand \newman {od()m}%
  {
    \IfValueTF{#2}
    {\begin{lateximage}[\textbackslash{}newman(#2)\{#3\}]}
    {\begin{lateximage}[\textbackslash{}newman\{#3\}]}
    \group_begin:
      \IfNoValueF  {#1} { \chemmacros_set_keys:nn {newman} {#1} }
      \IfNoValueTF {#2}
        { \chemmacros_newman:nn {  } {#3} }
        { \chemmacros_newman:nn {#2} {#3} }
    \group_end:
    \end{lateximage}
  }%

}{}% \@ifchemmacrosmoduleloaded
```

```
}% AtBeginDocument
```

§ 155.12 Orbital

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{orbital}{
\PackageInfo{lwarp}{Patching~chemmacros~module~orbital}

\RenewDocumentCommand \orbital {om}
  {
    \IfValueTF{#1}
    {
        \begin{lateximage}[%
            \textbackslash{}orbital{[}\LWR@HTMLsanitize{#1}{]}\{#2\}%
        ][][margin-left: 1em ; margin-right: 1em]
    }
    {
        \begin{lateximage}[%
            \textbackslash{}orbital\{#2\}%
        ][][margin-left: 1em ; margin-right: 1em]
    }
    \group_begin:
      \chemmacros_set_keys:nn {orbital/type} {#2}
      \IfNoValueTF {#1}
        { \chemmacros_orbital:n {  } }
        { \chemmacros_orbital:n {#1} }
    \group_end:
    \end{lateximage}
  }

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.13 Reactions

`\chemmacros_declare_reaction_env` {⟨*chem*⟩} {⟨*math*⟩} {⟨*args number*⟩} {⟨*argument list ({#2}{#3}...)*⟩}

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{reactions}{
\PackageInfo{lwarp}{Patching~chemmacros~module~reactions}

\cs_gset_protected:Npn \chemmacros_declare_reaction_env:nnnn #1#2#3#4
  {
    \exp_args:Nnx \DeclareDocumentEnvironment {#1} { O{} \prg_replicate:nn {#3+0} {m} }
      {
        \boolfalse{mathjax}%    lwarp
        \chemmacros_add_reaction_description:n {##1}
        \__chemmacros_begin_reaction:
        \chemmacros_reaction_read:nnw {#2} {#4}
```

```
      }
      {
        \__chemmacros_end_reaction:
      }
  }
\cs_generate_variant:Nn \chemmacros_declare_reaction_env:nnnn {nnnV}

\RenewChemReaction {reaction}   {equation}
\RenewChemReaction {reaction*}  {equation*}
\RenewChemReaction {reactions}  {align}
\RenewChemReaction {reactions*} {align*}

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.14 Redox

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{redox}{
\PackageInfo{lwarp}{Patching~chemmacros~module~redox}

\NewDocumentCommand \LWR@chemmacros@ox { s m >{\SplitArgument{1}{,}}m }
  {
    \IfBooleanTF {#1}
      { \chemmacros_ox:nnnn {#1} {#2} #3 }
      { \chemmacros_ox:nnnn {  } {#2} #3 }
  }

\RenewDocumentCommand \ox { s O{} m }
  {
    \begingroup
    \boolfalse{mathjax}
    \IfBooleanTF {#1}
      {
        \LWR@subsingledollar*{% yes hash
            \textbackslash{}ox*\{\LWR@HTMLsanitize{#3}\}% alt
        }{%
            star \protect\LWR@HTMLsanitize{\detokenize\expandafter{#2}}%
        }{%
            \LWR@chemmacros@ox* {#2} {#3}% contents
        }%
      }
      {
        \LWR@subsingledollar*{% yes hash
            \textbackslash{}ox*\{\LWR@HTMLsanitize{#3}\}% alt
        }{%
            \protect\LWR@HTMLsanitize{\detokenize\expandafter{#2}}%
        }{%
            \LWR@chemmacros@ox {#2} {#3}% contents
        }%
```

```
      }
    \endgroup
  }

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.15 Scheme

Fix for **chemmacros** as of v5.8b, when using **newfloat** and **babel**:

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{scheme}{
\PackageInfo{lwarp}{Patching~chemmacros~module~scheme}

\ifdefstring{\schemename}{los}{
\SetupFloatingEnvironment{scheme}{
name = \chemmacros_translate:n {scheme-name}
}
}{}

}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument
```

§ 155.16 Spectroscopy

```
\AtBeginDocument{
\@ifchemmacrosmoduleloaded{spectroscopy}{
\PackageInfo{lwarp}{Patching~chemmacros~module~spectroscopy}

\ChemCompatibilityTo{5.8}
\cs_gset_protected:Npn \__chemmacros_nmr_base:nn #1#2
  {
    \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
      {
        \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { \{ }
        \tl_put_right:Nn \g__chemmacros_nmr_element_coupled_tl { \} }
      }
    \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl {#2}
%     \chemmacros_chemformula:n { ^{#1} }
    \textsuperscript{#1}
    \bool_if:NTF \l__chemmacros_nmr_parse_bool
      { \chemformula_ch:nV {} \g__chemmacros_nmr_element_coupled_tl }
      { \chemmacros_chemformula:V \g__chemmacros_nmr_element_coupled_tl }
    \tl_use:N \l__chemmacros_nmr_element_method_connector_tl
    \tl_use:N \l__chemmacros_nmr_method_tl
  }
\EndChemCompatibility
\ChemCompatibilityFrom{5.8}
```

```
\cs_gset_protected:Npn \__chemmacros_nmr_base:nn #1#2
  {
    \group_begin:
      \tl_use:N \l__chemmacros_nmr_base_format_tl
      \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
        {
          \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { \{ }
          \tl_put_right:Nn \g__chemmacros_nmr_element_coupled_tl { \} }
        }
      \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl {#2}
%       \chemmacros_chemformula:n { ^{#1} }
      \textsuperscript{#1}
      \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
        {
          \bool_if:NTF \l__chemmacros_nmr_parse_bool
            { \chemformula_ch:nV {} \g__chemmacros_nmr_element_coupled_tl }
            { \chemmacros_chemformula:V \g__chemmacros_nmr_element_coupled_tl }
        }
      \tl_use:N \l__chemmacros_nmr_element_method_connector_tl
      \tl_use:N \l__chemmacros_nmr_method_tl
    \group_end:
  }
\EndChemCompatibility


\cs_gset_protected:Npn \chemmacros_nmr_position:n #1
  {
    \chemmacros_chemformula:x
      {
        \exp_not:V \g__chemmacros_nmr_element_tl
        \bool_if:NF \l__chemmacros_nmr_position_side_bool
          {
            \tl_if_eq:NnTF \l__chemmacros_nmr_position_tl {^}% lwarp
            { \textsuperscript{\exp_not:n { {#1} }} }% lwarp
            { \textsubscript{\exp_not:n { {#1} }} }% lwarp
%             \exp_not:V \l__chemmacros_nmr_position_tl
%             \exp_not:n { {#1} }
          }
      }
    \bool_if:NT \l__chemmacros_nmr_position_side_bool
      {
        \tl_use:N \l__chemmacros_nmr_position_tl
        \__chemmacros_nmr_position:n {#1}
      }
  }

\cs_gset_protected:Npn \__chemmacros_nmr_coupling:w (#1;#2)
  {
    \tl_set:Nn \l__chemmacros_nmr_coupling_bonds_tl
      {
```

```
        \l__chemmacros_nmr_coupling_bonds_pre_tl
        #1
        \l__chemmacros_nmr_coupling_bonds_post_tl
      }
    \bool_if:NTF \l__chemmacros_nmr_coupling_nuclei_sub_bool
      {
        \tl_set:Nn \l__chemmacros_nmr_coupling_nuclei_tl
          {
%             \c_math_subscript_token
            \textsubscript% lwarp
              {
                \l__chemmacros_nmr_coupling_nuclei_pre_tl
                \chemmacros_chemformula:n {#2}
                \l__chemmacros_nmr_coupling_nuclei_post_tl
              }
          }
      }
      {
        \tl_set:Nn \l__chemmacros_nmr_coupling_nuclei_tl
          {
            \l__chemmacros_nmr_coupling_nuclei_pre_tl
            \chemmacros_chemformula:n {#2}
            \l__chemmacros_nmr_coupling_nuclei_post_tl
          }
      }
    \__chemmacros_nmr_coupling_aux_i:w
  }

\AfterEndPreamble{% After \AtBeginDocument
% \NMR{<num>,<elem>}(<num>,<unit>)[<solvent>] ALL arguments are optional
% \NMR* same but without ": $\delta$" at end
\cs_gset_protected:Npn \chemmacros_nmr:nnnn #1#2#3#4
    {
    \bool_if:NT \l__chemmacros_nmr_list_bool { \item \scan_stop: }
    \group_begin:
        \chemmacros_leave_vmode:
        \bool_set_false:N \l__chemmacros_nmr_frequency_bool
        \bool_set_false:N \l__chemmacros_nmr_solvent_bool
        \tl_if_empty:nF {#3}
        { \bool_set_true:N \l__chemmacros_nmr_frequency_bool }
        \tl_if_empty:nF {#4}
        { \bool_set_true:N \l__chemmacros_nmr_solvent_bool }
        \bool_if:nT
        {
            \l__chemmacros_nmr_frequency_bool
            ||
            \l__chemmacros_nmr_solvent_bool
        }
        { \bool_set_true:N \l__chemmacros_nmr_delimiters_bool }
        \bool_if:nT
```

```
        {
            \l__chemmacros_nmr_frequency_bool
            &&
            \l__chemmacros_nmr_solvent_bool
        }
        { \bool_set_true:N \l__chemmacros_nmr_comma_bool }
        \tl_if_empty:nTF {#2}
        {
            \__chemmacros_nmr_nucleus:VV
            \l__chemmacros_nmr_isotope_default_tl
            \l__chemmacros_nmr_element_default_tl
        }
        { \__chemmacros_nmr_nucleus:w #2 \q_stop }
        \mode_if_math:TF
        {
            \text
            {
                \group_begin:
                \tl_use:N \l__chemmacros_nmr_format_tl
\LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
                \__chemmacros_nmr_base:VV
                    \g__chemmacros_nmr_isotope_tl
                    \g__chemmacros_nmr_element_tl
                \bool_if:NT \l__chemmacros_nmr_delimiters_bool
                    { ~ ( }
                \bool_if:NT \l__chemmacros_nmr_frequency_bool
                    { \__chemmacros_nmr_frequency:n {#3} }
                \bool_if:NT \l__chemmacros_nmr_comma_bool
                    { , ~ }
                \bool_if:NT \l__chemmacros_nmr_solvent_bool
                    { \chemmacros_chemformula:n {#4} }
                \bool_if:NT \l__chemmacros_nmr_delimiters_bool
                    { ) }
                \tl_if_blank:nT {#1} {:~}
}}% lwarp
                \group_end:
            }
            \tl_if_blank:nT {#1}
            {
                \delta
                \text { \l__chemmacros_nmr_delta_tl }
                \bool_if:NT \l__chemmacros_nmr_use_equal_bool {=}
            }
        }
        {
            \group_begin:
            \tl_use:N \l__chemmacros_nmr_format_tl
\LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
            \__chemmacros_nmr_base:VV
                \g__chemmacros_nmr_isotope_tl
```

```
                \g__chemmacros_nmr_element_tl
            \bool_if:NT \l__chemmacros_nmr_delimiters_bool
                {~(}
            \bool_if:NT \l__chemmacros_nmr_frequency_bool
                { \__chemmacros_nmr_frequency:n {#3} }
            \bool_if:NT \l__chemmacros_nmr_comma_bool
                {,~}
            \bool_if:NT \l__chemmacros_nmr_solvent_bool
                {
                \bool_if:NTF \l__chemmacros_nmr_parse_bool

%                     { \chemformula_ch:nn { } {#4} }%  original
                    {\ch{#4}}%  lwarp
                    {#4}
                }
            \bool_if:NT \l__chemmacros_nmr_delimiters_bool
                {)}
}}% lwarp
            \tl_if_blank:nT {#1} {:}
            \group_end:
            \tl_if_blank:nT {#1}
            {
                \tl_use:N \c_space_tl
                \c_math_toggle_token
                \delta
                \c_math_toggle_token
                \l__chemmacros_nmr_delta_tl
                \bool_if:NT \l__chemmacros_nmr_use_equal_bool {~=}
            }
        }
    \group_end:
    }
}% AfterEndPremble


\RenewDocumentCommand \chemmacros_data:w { smo }
  {
    \bool_if:NT \l__chemmacros_nmr_list_bool { \item }
      {
%         \tl_use:N \l__chemmacros_nmr_format_tl #2
        \tl_use:N \l__chemmacros_nmr_format_tl
        \LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
            #2
            \IfNoValueF {#3} { ~ ( #3 ) }
            \IfBooleanT {#1} { \bool_if:NT \l__chemmacros_nmr_use_equal_bool { : } }
        }}% lwarp
      }
    \IfBooleanF {#1} { \bool_if:NT \l__chemmacros_nmr_use_equal_bool { ~ = } }
  }
```

```
765
766 }{}% \@ifchemmacrosmoduleloaded
767 }% AtBeginDocument
```

§ 155.17 Thermodynamics

```
768 \AtBeginDocument{
769 \@ifchemmacrosmoduleloaded{thermodynamics}{
770 \PackageInfo{lwarp}{Patching~chemmacros~module~thermodynamics}
771
772 \cs_gset_protected:Npn \chemmacros_state:nn #1#2
773  {
774    \group_begin:
775      \boolfalse{mathjax}
776      \chemmacros_set_keys:nn {thermodynamics} {#1}
777        \LWR@subsingledollar*{% yes hashing
778            \textbackslash{}state\{\LWR@HTMLsanitize{#2}\}% alt
779        }{%
780            chemmacros_state% add'l hashing
781            #1% options
782            LSP \tl_use:N \l__chemmacros_state_sp_left_tl% super/subscripts
783            LSB \tl_use:N \l__chemmacros_state_sb_left_tl
784            RSP \tl_use:N \l__chemmacros_state_sp_right_tl
785            RSB \tl_use:N \l__chemmacros_state_sb_right_tl
786        }
787        {
788         \LWR@origensuredmath{
789          \chemmacros_text:V \l__chemmacros_state_pre_tl
790          \c_math_superscript_token
791            { \chemmacros_text:V \l__chemmacros_state_sp_left_tl }
```

Only add the subscripts if they are being used. This avoids causing an incorrect depth, as the empty subscript will be measured by TEX but cropped out by **pdfcrop**.

```
792          \tl_if_empty:NTF \l__chemmacros_state_sb_left_tl
793          {}
794          {
795            \c_math_subscript_token
796            { \chemmacros_text:V \l__chemmacros_state_sb_left_tl }
797          }
798          #2
799          \c_math_superscript_token
800            { \chemmacros_text:V \l__chemmacros_state_sp_right_tl }
801          \tl_if_empty:NTF \l__chemmacros_state_sb_right_tl
802          {}
803          {
804            \c_math_subscript_token
805            { \chemmacros_text:V \l__chemmacros_state_sb_right_tl }
806          }
```

```
          \chemmacros_text:V \l__chemmacros_state_post_tl
        }
      }
    \group_end:
  }
\cs_generate_variant:Nn \chemmacros_state:nn { nV }

\cs_gset_protected:Npn \chemmacros_declare_state:Nn #1#2
  {
    \chemmacros_define_keys:xn
      {thermodynamics/\chemmacros_remove_backslash:N #1}
      {
        pre               .meta:nn = {chemmacros/thermodynamics} { pre = ##1 } ,
        post              .meta:nn = {chemmacros/thermodynamics} { post = ##1 } ,
        superscript-left  .meta:nn = {chemmacros/thermodynamics} { superscript-left = ##1 } ,
        superscript-right .meta:nn = {chemmacros/thermodynamics} { superscript-right = ##1 } ,
        superscript       .meta:n  = { superscript-right = ##1 } ,
        subscript-left    .meta:nn = {chemmacros/thermodynamics} { subscript-left = ##1 } ,
        subscript-right   .meta:nn = {chemmacros/thermodynamics} { subscript-right = ##1 } ,
        subscript         .meta:n     = { subscript-left = ##1 } ,
        subscript-pos     .choices:nn =
          { left , right }
          { \tl_set_eq:NN \l__chemmacros_state_sb_pos_tl \l_keys_choice_tl } ,
        symbol            .tl_set:N = \l__chemmacros_state_symbol_tl ,
        unit              .tl_set:N = \l__chemmacros_state_unit_tl
      }
    \DeclareDocumentCommand #1 { sO{}D(){}m }
      {
        \group_begin:
          \chemmacros_set_keys:xn
            {thermodynamics/\chemmacros_remove_backslash:N #1}
            {#2}
          \tl_if_blank:nF {##3}
            {
              \chemmacros_set_keys:nx {thermodynamics}
                { subscript-\l__chemmacros_state_sb_pos_tl = \exp_not:n {##3} }
            }
              \chemmacros_state:nV {##2} \l__chemmacros_state_symbol_tl
              \chemmacros_set_keys_groups:nnn {thermodynamics} {variables} {##2}
              \IfBooleanF {##1} {  = ~ \SI {##4} { \l__chemmacros_state_unit_tl } }
        \group_end:
      }
  }
```

The pre-existing macros are redefined with the new definition:

```
\RenewChemState \enthalpy { symbol = H , unit = \kilo\joule\per\mole }
\RenewChemState \entropy  { symbol = S , unit = \joule\per\kelvin\per\mole , pre = }
\RenewChemState \gibbs    { symbol = G , unit = \kilo\joule\per\mole }
```

```
}{}% \@ifchemmacrosmoduleloaded
}% AtBeginDocument

\ExplSyntaxOff
```

File 64 `lwarp-chemnum.sty`

§ 156 Package chemnum

(*Emulates or patches code by* Clemens Niederberger.)

Pkg `chemnum` **chemnum** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{chemnum}

\ExplSyntaxOn

\cs_gset_protected:Npn \chemnum_compound_write:n #1
  {
    \chemnum_get_compound_property:nn {#1} {pre-main-label-code}
    \group_begin:
      \bool_if:NTF \l__chemnum_compound_local_bool
        { \l__chemnum_local_label_format_tl }
        { \chemnum_get_compound_property:nn {#1} {label-format} }
      {
        \LWR@textcurrentfont{
            \chemnum_get_compound_property:nn {#1} {counter-representation}
        }
      }
    \group_end:
    \chemnum_get_compound_property:nn {#1} {post-main-label-code}
  }

\cs_gset_protected:Npn \chemnum_subcompound_write:nn #1#2
  {
    \group_begin:
      \bool_if:NTF \l__chemnum_compound_local_bool
        { \l__chemnum_local_label_format_tl }
        { \chemnum_get_compound_property:nn {#1} {label-format} }
      {
        \LWR@textcurrentfont{
            \chemnum_get_subcompound_property:nnn {#1} {#2}
            {counter-representation}
        }
      }
    \group_end:
  }
```

```
34
35 \ExplSyntaxOff
```

File 65 `lwarp-cite.sty`

§ 157 Package cite

(*Emulates or patches code by* Donald Arseneau.)

Pkg `cite` **cite** is patched for use by **lwarp**.

for HTML output:

```
1 \LWR@ProvidesPackagePass{cite}
```

For the [super] option, the `\kern` must be removed:

```
2 \def\LWRCT@biblabel#1{\@citess{#1}\kern-\labelsep\,}
3
4 \ifdefstrequal{\@biblabel}{\LWRCT@biblabel}
5 {
6     \def\@biblabel#1{\@citess{#1}}
7 }{}
```

For the [super] option, `\textsuperscript` is used instead of math superscript:

```
8 \def\@citess#1{\textsuperscript{#1}}
9
10 \DeclareDocumentCommand\citepunct{}{,\,\relax}
```

File 66 `lwarp-clrdblpg.sty`

§ 158 Package clrdblpg

Pkg `clrdblpg` **clrdblpg** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{clrdblpg}
```

File 67 `lwarp-color.sty`

§ 159 Package color

Pkg `color` Allowed but ignored. **xcolor** is then required as well.

color is superceded by **xcolor**, and **lwarp** requires several of the features of **xcolor**.

⚠ missing colors It should be sufficient for the user's document to load **color** then load **xcolor** as well.

for HTML output:

```
\LWR@ProvidesPackagePass{color}
\RequirePackage{xcolor}
```

File 68 `lwarp-colortbl.sty`

§ 160 Package

colortbl

Pkg colortbl **colortbl** is emulated.

⚠ row/cell color Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

colortbl ignores the overhang arguments.

for HTML output: A few placeholder definitions are forgotten first:

```
\let\rowcolor\relax

\LWR@ProvidesPackagePass{colortbl}
```

The following `\LWR@HTML` versions are used inside an HTML `tabular`.

`\columncolor` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}` `[`⟨*left overhang*⟩`]` `[`⟨*right overhang*⟩`]`

`\LWR@getmynexttoken` is not used here because `\columncolor` is not used inside the data area of the tabular.

```
\NewDocumentCommand{\LWR@HTML@columncolor}{O{named} m o o}{%
\convertcolorspec{#1}{#2}{HTML}\LWR@columnHTMLcolor%
\LWR@addtabularcellcolor%
}

\LWR@formatted{columncolor}
```

`\LWR@getmynexttoken` is used for `\rowcolor` because it is used inside the data area of the tabular.

`\rowcolor` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}` `[`⟨*left overhang*⟩`]` `[`⟨*right overhang*⟩`]`

```
\NewDocumentCommand{\LWR@HTML@rowcolor}{O{named} m o o}{%
\convertcolorspec{#1}{#2}{HTML}\LWR@rowHTMLcolor%
\LWR@getmynexttoken%
}

```

```
\LWR@expandableformatted{rowcolor}
```

`\cellcolor` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}` `[`⟨*left overhang*⟩`]` `[`⟨*right overhang*⟩`]`

```
\NewDocumentCommand{\LWR@HTML@cellcolor}{O{named} m o o}{%
\convertcolorspec{#1}{#2}{HTML}\LWR@cellHTMLcolor%
\LWR@addtabularcellcolor%
}

\LWR@formatted{cellcolor}
```

`\arrayrulecolor` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}`

The HTML version for use outside a `tabular`. Inside a `tabular`, `\LWR@HTML@arrayrulecolornexttoken` is used instead.

```
\newcommand{\LWR@HTML@arrayrulecolor}[2][named]{%
\convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
}

\LWR@expandableformatted{arrayrulecolor}
```

`\LWR@arrayrulecolornexttoken` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}`

The HTML version for use inside a `tabular`.

```
\newcommand{\LWR@HTML@arrayrulecolornexttoken}[2][named]{%
\convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
\LWR@getmynexttoken%
}

\LWR@expandableformatted{arrayrulecolornexttoken}
```

`\doublerulesepcolor` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}`

The version for use outside a `tabular`.

```
\newcommand{\LWR@HTML@doublerulesepcolor}[2][named]{}

\LWR@expandableformatted{doublerulesepcolor}
```

`\LWR@doublerulesepcolornexttoken` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}`

The version for use inside a `tabular`.

```
\newcommand{\LWR@HTML@doublerulesepcolornexttoken}[2][named]{\LWR@getmynexttoken}

\LWR@expandableformatted{doublerulesepcolornexttoken}
```

File 69 `lwarp-continue.sty`

§ 161 Package continue

Pkg `continue` **continue** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{continue}

\newcommand*{\flagcont}{}
\newcommand*{\flagend}{}
\newcommand*{\flagword}{}
\newcommand*{\preflagword}{}
\newcommand*{\postflagword}{}
\newlength\contsep
\newlength\contdrop
```

File 70 `lwarp-crop.sty`

§ 162 Package crop

(*Emulates or patches code by* Melchior FRANZ.)

Pkg `crop` Emulated.

for HTML output: Discard all options for **lwarp-crop**:

```
\LWR@ProvidesPackageDrop{crop}

\newcommand*{\crop}[1][]{}
\newcommand*{\cropdef}[6][]{}
```

File 71 `lwarp-cuted.sty`

§ 163 Package cuted

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `cuted` **cuted** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{cuted}
```

```
\newenvironment{strip}{}{}
\newskip\stripsep
\def\oldcolsbreak#1{}
```

File 72 `lwarp-cutwin.sty`

§ 164 Package cutwin

(*Emulates or patches code by* Peter Wilson and Alan Hoenig.)

Pkg `cutwin` Emulated.

for HTML output: Discard all options for **lwarp-cutwin**:

```
\LWR@ProvidesPackageDrop{cutwin}

\newcommand*{\opencutleft}{}
\newcommand*{\opencutright}{}
\newcommand*{\opencutcenter}{}
\newcommand*{\cutfuzz}{}

\newenvironment{cutout}[4]
{\marginpar{\windowpagestuff}}
{}

\newcommand*{\windowpagestuff}{}

\newcommand*{\pageinwindow}{%
% \begin{minipage}{.3\linewidth}
\windowpagestuff
% \end{minipage}
}

\newenvironment{shapedcutout}[3]
{\marginpar{\picinwindow}}
{}

\newcommand*{\putstuffinpic}{}

\newcommand*{\picinwindow}{%
\begin{picture}(0,0)
\putstuffinpic
\end{picture}}
```

File 73 `lwarp-dblfloatfix.sty`

§ 165 Package dblfloatfix

Pkg `dblfloatfix` **dblfloatfix** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{dblfloatfix}
```

File 74 `lwarp-dblfnote.sty`

§ 166 Package dblfnote

(*Emulates or patches code by* Hiroshi Nakashima.)

Pkg `dblfnote` **dblfnote** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{dblfnote}

\newcounter{DFNsloppiness}
\newdimen\DFNcolumnsep
\newdimen\DFNcolumnwidth
\def\DFNallowcbreak{}
\def\DFNinhibitcbreak{}
\def\DFNtrysingle{}
\def\DFNalwaysdouble{}
\def\DFNruleboth{}
\def\DFNruleleft{}
```

File 75 `lwarp-dcolumn.sty`

§ 167 Package dcolumn

Pkg `dcolumn` **dcolumn** is emulated by the **lwarp** core.

```
\LWR@ProvidesPackageDrop{dcolumn}
```

File 76 `lwarp-diagbox.sty`

§ 168 Package

diagbox

(*Emulates or patches code by* Leo Liu.)

Pkg `diagbox` **diagbox** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{diagbox}
```

To restore print-mode inside a `lateximage`:

```
\LetLtxMacro\LWR@origdiagbox@double\diagbox@double
\LetLtxMacro\LWR@origdiagbox@triple\diagbox@triple

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\diagbox@double\LWR@origdiagbox@double%
\LetLtxMacro\diagbox@triple\LWR@origdiagbox@triple%
}
```

`\LWR@diagbox@AB` {⟨*E/W*⟩} {⟨*A*⟩} {⟨*E/W*⟩} {⟨*B*⟩}

```
\newcommand{\LWR@diagbox@AB}[4]{
\begingroup%
\LetLtxMacro\\\newline%
\BlockClassSingle{diagbox#1}{#2}%
\BlockClassSingle{diagbox#3}{#4}%
\endgroup%
\LWR@stoppars%
}
```

`\LWR@diagboxNW` {⟨*A*⟩} {⟨*B*⟩}

```
\newcommand{\LWR@diagboxNW}[2]{%
\LWR@diagbox@AB{E}{#2}{W}{#1}%
}
```

Likewise for NE, SW, SE:

```
\newcommand{\LWR@diagboxNE}[2]{%
\LWR@diagbox@AB{W}{#1}{E}{#2}%
}

\let\LWR@diagboxSW\LWR@diagboxNE
\let\LWR@diagboxSE\LWR@diagboxNW
```

`\diagbox@double` {⟨*keys*⟩} {⟨*A*⟩} {⟨*B*⟩}

```
\def\diagbox@double#1#2#3{%
\setkeys{diagbox}{dir=NW,#1}%
\@nameuse{LWR@diagbox\diagbox@dir}{#2}{#3}%
}
```

`\LWR@diagboxTNW` {⟨*title*⟩} {⟨*A*⟩} {⟨*B*⟩}

```
\newcommand{\LWR@diagboxTNW}[3]{%
\BlockClassSingle{diagboxtitleN}{#1}
\LWR@diagboxNW{#2}{#3}
}
```

Likewise for NE, SW, SE:

```
\newcommand{\LWR@diagboxTNE}[3]{%
\BlockClassSingle{diagboxtitleN}{#1}
\LWR@diagboxNE{#2}{#3}
}

\newcommand{\LWR@diagboxTSW}[3]{%
\LWR@diagboxSW{#2}{#3}
\BlockClassSingle{diagboxtitleS}{#1}
\LWR@stoppars%
}

\newcommand{\LWR@diagboxTSE}[3]{%
\LWR@diagboxSE{#2}{#3}
\BlockClassSingle{diagboxtitleS}{#1}
\LWR@stoppars%
}
```

`\diagbox@triple` {⟨*keys*⟩} {⟨*A*⟩} {⟨*T*⟩} {⟨*B*⟩}

```
\def\diagbox@triple#1#2#3#4{%
\setkeys{diagbox}{dir=NW,#1}%
\@nameuse{LWR@diagboxT\diagbox@dir}{#3}{#2}{#4}%
}
```

File 77 `lwarp-draftcopy.sty`

§ 169 Package draftcopy

Pkg `draftcopy` **draftcopy** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{draftcopy}

\newcommand{\draftcopyVersion}[1]{}
\newcommand{\draftcopySetGrey}[1]{}
\newcommand{\draftcopySetScale}[1]{}
\newcommand{\draftcopySetScaleFactor}[1]{}
\newcommand{\draftcopyFirstPage}[1]{}
\newcommand{\draftcopyLastPage}[1]{}
\newcommand{\draftcopyName}[2]{}
\newcommand{\draftcopyPageTransform}[1]{}
\newcommand{\draftcopyBottomTransform}[1]{}
\newcommand{\draftcopyPageX}[1]{}
\newcommand{\draftcopyPageY}[1]{}
\newcommand{\draftcopyBottomX}[1]{}
\newcommand{\draftcopyBottomY}[1]{}
```

File 78 `lwarp-draftfigure.sty`

§ 170 Package draftfigure

Pkg `draftfigure` **draftfigure** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{draftfigure}[2017/07/19]
\RequirePackage{xkeyval}

\define@key{draftfigure}{code}{}
\define@key{draftfigure}{noframe}[true]{}
\define@key{draftfigure}{filename}[true]{}
\define@key{draftfigure}{content}[]{}
\define@key{draftfigure}{style}[normal]{}
\define@key{draftfigure}{position}[left]{}
\define@key{draftfigure}{size}[normal]{}
\newcommand\setdf[1]{\setkeys{draftfigure}{#1}}
```

File 79 `lwarp-draftwatermark.sty`

§ 171 Package draftwatermark

(*Emulates or patches code by* Sergio Callegari.)

Pkg `draftwatermark` **draftwatermark** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{draftwatermark}
```

```
\newcommand{\SetWatermarkAngle}[1]{}
\newcommand{\SetWatermarkColor}[1]{}
\newcommand{\SetWatermarkLightness}[1]{}
\newcommand{\SetWatermarkFontSize}[1]{}
\newcommand{\SetWatermarkScale}[1]{}
\newcommand{\SetWatermarkHorCenter}[1]{}
\newcommand{\SetWatermarkVertCenter}[1]{}
\newcommand{\SetWatermarkText}[1]{}
```

File 80 `lwarp-easy-todo.sty`

§ 172 Package easy-todo

(*Emulates or patches code by* Juan Rada-Vilela.)

Pkg `easy-todo` **easy-todo** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{easy-todo}
```

`\listoftodos` Modified to correct buggy use of `\flushright`.

```
\let\LWR@origlistoftodos\listoftodos

\renewcommand{\listoftodos}{%
\begingroup
\renewcommand{\flushright}{}
\LWR@origlistoftodos
\endgroup
}
```

`\todoii` Modified to use `\textcolor` instead of `\color`.

```
\renewcommand{\todoii}[2]{%
\ifthenelse{\equal{\@todoobeyfinal}{true}}{%
    \ifoptionfinal{\todoenable{false}}{\todoenable{true}}%
}{}%
\ifthenelse{\equal{\@todoenable}{true}}{%
\refstepcounter{todos}%
\noindent{%
    \todocolor%
    \LWR@textcurrentcolor{%
        \normalfont\scriptsize{\bfseries{\thetodos.#1}}%
    }%
}%
\addcontentsline{lod}{todos}{\protect{\thetodos. }#2}%
}{}%
```

```
24 }
```

File 81 `lwarp-ebook.sty`

§ 173 Package ebook

(*Emulates or patches code by* Jørgen Steensgaard.)

Pkg `ebook` **ebook** is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{ebook}

2 \setcounter{secnumdepth}{0}
3 \setcounter{tocdepth}{2}
4
5 \providecommand{\pagefill}[1][0.001mm]{\noindent}
6
7 \providecommand{\ebook}{
8 \setcounter{secnumdepth}{0}
9 \setcounter{tocdepth}{2}
10 }
```

File 82 `lwarp-ellipsis.sty`

§ 174 Package ellipsis

(*Emulates or patches code by* Peter J. Heslin.)

Pkg `ellipsis` **ellipsis** is emulated.

```
1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}
4
5 \newcommand*{\midwordellipsis}{\,\textellipsis\,}
```

File 83 `lwarp-emptypage.sty`

§ 175 Package emptypage

Pkg `emptypage` **emptypage** is ignored.

for HTML output: Discard all options for **lwarp-emptypage**:

```
\LWR@ProvidesPackageDrop{emptypage}
```

File 84 `lwarp-endfloat.sty`

§ 176 Package endfloat

Pkg `endfloat` **endfloat** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{endfloat}

\newcommand\figureplace{}
\newcommand\tableplace{}
\newcommand\floatplace[1]{}
\newcounter{posttable}
\newcounter{postfigure}
\newcommand*{\thepostbl}{}
\newcommand*{\thepostfig}{}
\newcommand{\AtBeginFigures}[1]{}
\newcommand{\AtBeginTables}[1]{}
\newcommand{\AtBeginDelayedFloats}[1]{}
\newcommand*{\processdelayedfloats}{}
\newcommand*{\efloatseparator}{}
\def\efloattype{}
\providecommand\efloatheading[1]{}
\providecommand\efloatpreamble{}
\providecommand\efloatpostamble{}
```

File 85 `lwarp-endheads.sty`

§ 177 Package endheads

Pkg `endheads` **endheads** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{endheads}

\newcommand{\changesinglepageabbrev}[1]{}
\newcommand{\changemultiplepageabbrev}[1]{}
\newcommand{\changenotesname}[1]{}
\newcommand{\changenotesheader}[1]{}
\newcommand{\changenotescontentsname}[1]{}
\newcommand{\changechapternotesline}[1]{}
\newcommand{\checknoteheaders}{}
```

```
\newif\ifnotesincontentson \notesincontentsonfalse
\newcommand{\notesincontents}{\notesincontentsontrue}
\newif\ifendnoteheaderson \endnoteheadersonfalse
\newcommand{\setupendnoteheaders}{%
    \endnoteheadersontrue%
}
\newif\iftitleinnotes \titleinnotestrue
\newcommand{\styleforchapternotebegin}{}
\newcommand{\styleforchapternoteend}{}
\newcommand{\setstyleforchapternotebegin}[1]{%
    \renewcommand{\styleforchapternotebegin}{#1}%
}
\newcommand{\setstyleforchapternoteend}[1]{%
    \renewcommand{\styleforchapternoteend}{#1}%
}
\newcommand{\resetendnotes}{}
\newif\ifnotesbychapteron \notesbychapteronfalse
\newcommand{\notesbychapter}{\notesbychapterontrue}
```

File 86 `lwarp-endnotes.sty`

§ 178 Package endnotes

(*Emulates or patches code by* John Lavagnino.)

Pkg `endnotes` Used as-is.

table of contents To place the endnotes in the TOC, use:

```
\usepackage{endnotes}
\appto\enoteheading{\addcontentsline{toc}{section}{\notesname}}
\renewcommand*{\notesname}{Endnotes} % optional
```

HTML page To additionally have the endnotes on their own HTML page, if `FileDepth` allows:

```
\ForceHTMLPage
\theendnotes
```

for HTML output:

```
\LWR@ProvidesPackagePass{endnotes}
```

```
\def\enoteformat{%
% \rightskip\z@ \leftskip\z@ \parindent=1.8em
\leavevmode
% \llap{
\makeenmark
% }
```

```
}

\def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
\def\makeenmark{\@makeenmark}
```

File 87 `lwarp-enumerate.sty`

§ 179 Package enumerate

Pkg `enumerate` **enumerate** is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by **lwarp** and thus must no longer be dropped.

for HTML output:

```
\LWR@ProvidesPackagePass{enumerate}
```

File 88 `lwarp-enumitem.sty`

§ 180 Package enumitem

(*Emulates or patches code by* Javier Bezos.)

Pkg `enumitem` **enumitem** is supported with minor adjustments.

for HTML output:

```
\LWR@ProvidesPackagePass{enumitem}
```

for HTML output:

```
\begin{warpHTML}
```

`\newlist` {⟨*name*⟩} {⟨*type*⟩} {⟨*maxdepth*⟩}
`\renewlist` {⟨*name*⟩} {⟨*type*⟩} {⟨*maxdepth*⟩}

For **enumitem** lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```
\let\LWR@orignewlist\newlist

\renewcommand*{\newlist}[3]{%
\LWR@orignewlist{#1}{#2}{#3}%
\AtBeginEnvironment{#1}{\@nameuse{LWR@#2start}}%
\AtEndEnvironment{#1}{\@nameuse{LWR@#2end}}%
}
```

```
\end{warpHTML}
```

File 89 `lwarp-epigraph.sty`

§ 181 Package epigraph

(*Emulates or patches code by* Peter Wilson.)

Pkg epigraph **epigraph** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{epigraph}

\DeclareDocumentCommand{\qitem}{m m}
{
\begin{BlockClass}{qitem}
#1
\ifbool{FormatWP}
{\begin{BlockClass}[border-top:1px solid gray]{epigraphsource}}
{\begin{BlockClass}{epigraphsource}}
#2
\end{BlockClass}
\end{BlockClass}
}

\DeclareDocumentCommand{\epigraph}{m m}
{
\begin{LWR@BlockClassWP}{\LWR@print@mbox{text-align:right}}{}{epigraph}
\qitem{#1}{#2}
\end{LWR@BlockClassWP}
}

\DeclareDocumentEnvironment{epigraphs}{}
{\LWR@BlockClassWP{\LWR@print@mbox{text-align:right}}{}{epigraph}}
{\endLWR@BlockClassWP}
```

Use css to format epigraphs.

The following are null commands for source compatibility:

```
\newenvironment*{flushepinormal}{}{}

\@ifclassloaded{memoir}{
\setlength{\epigraphwidth}{.5\linewidth}
\renewcommand{\textflush}{flushepinormal}
\renewcommand{\epigraphhead}[2][0]{#2}
```

```
\renewcommand{\dropchapter}[1]{}
\renewcommand*{\undodrop}{}
}{% not memoir
\newlength{\epigraphwidth}
\setlength{\epigraphwidth}{.5\linewidth}
\newcommand{\textflush}{flushepinormal}
\newcommand{\epigraphflush}{flushright}
\newcommand{\sourceflush}{flushright}
\newcommand*{\epigraphsize}{\small}
\newlength{\epigraphrule}
\newlength{\beforeepigraphskip}
\newlength{\afterepigraphskip}
\newcommand{\epigraphhead}[2][0]{#2}
\newcommand{\dropchapter}[1]{}
\newcommand*{\undodrop}{}
}% not memoir

\let\cleartoevenpage\relax% also in nextpage
\newcommand{\cleartoevenpage}[1][]{}
```

File 90 `lwarp-epstopdf.sty`

§ 182 Package epstopdf

Pkg epstopdf

Previous versions of **lwarp** had a nullfied verison, but now **epstopdf-base** is supported. **lwarp-epstopdf** becomes a placeholder to overwrite previous versions.

See package **epstopdf-base** for details.

for HTML output:

```
\LWR@ProvidesPackagePass{epstopdf}[2016/05/15]
```

File 91 `lwarp-epstopdf-base.sty`

§ 183 Package epstopdf-base

Pkg epstopdf-base

⚠ convert to .svg

Images with an `.eps` extension will be converted to `.pdf`. The HTML output uses the `.svg` version, so use

Enter ⇒ `lwarpmk pdftosvg <listofPDFfiles>`

to generate `.svg` versions.

for HTML output:

```
\LWR@ProvidesPackagePass{epstopdf-base}[2016/05/15]
```

Redefine to remember the image filename, replacing `.pdf` with `.svg`. Use the **epstopdf** print version inside a `lateximage`.

```
\newcommand*{\LWR@HTML@ETE@OrgGin@setfile}[3]{%
    \edef\LWR@tempone{#3}%
    \StrSubstitute{\LWR@tempone}{.pdf}{.svg}[\LWR@tempone]%
    \StrSubstitute{\LWR@tempone}{.PDF}{.SVG}[\LWR@tempone]%
    \xdef\LWR@parsedfilename{\LWR@tempone}%
}

\LWR@formatted{ETE@OrgGin@setfile}
```

`\includegraphics` in HTML mode redefines `\Gin@setfile` to be `\LWR@HTML@Gin@setfile`, which is now redirected to **epstopdf**'s version:

```
\renewcommand*{\LWR@HTML@Gin@setfile}[3]{%
    \ETE@Gin@setfile{#1}{#2}{#3}%
}
```

Allow `.eps` images to be found if a suffix is not provided:

```
\AtBeginDocument{
\DeclareGraphicsExtensions{%
    .eps,.EPS,.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG%
}
\DeclareGraphicsRule{.svg}{svg}{.svg}{}
\DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
}
```

Likewise when inside a `lateximage`:

```
\appto\LWR@restoreorigformatting{%
\DeclareGraphicsExtensions{%
    .eps,.EPS,.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG%
}%
}
```

File 92 `lwarp-errata.sty`

§ 184 Package errata

(*Emulates or patches code by* Michael Kohlhase.)

Pkg `errata` **errata** is patched for use by **lwarp**.

This is for v0.3 of **errata**. A newer version of **errata** with more features is under development, at which time the **lwarp** version will have to be updated.

for HTML output:

Macros are being defined with the math dollar, so enable the HTML version during package loading:

```
\StartDefiningMath
```

Now load the package:

```
\LWR@ProvidesPackagePass{errata}[2006/11/12]
```

Patches for dynamic inline math:

```
\xpatchcmd{\erratumAdd}
    {$_a^{\arabic{erratum}}$}
%     {\StartDynamicMath$_a^{\arabic{erratum}}$\StopDynamicMath}
    {\textsubscript{a}\textsuperscript{\arabic{erratum}}}
    {}
    {\LWR@patcherror{erratum}{erratumAdd}}

\xpatchcmd{\erratumDelete}
    {$_d^{\arabic{erratum}}$}
%     {\StartDynamicMath$_d^{\arabic{erratum}}$\StopDynamicMath}
    {\textsubscript{d}\textsuperscript{\arabic{erratum}}}
    {}
    {\LWR@patcherror{erratum}{erratumDelete}}

\xpatchcmd{\erratumReplace}
    {$_r^{\arabic{erratum}}$}
%     {\StartDynamicMath$_r^{\arabic{erratum}}$\StopDynamicMath}
    {\textsubscript{r}\textsuperscript{\arabic{erratum}}}
    {}
    {\LWR@patcherror{erratum}{erratumReplace}}

\xpatchcmd{\erratum}
    {$_a$}
%     {\StartDynamicMath$_a$\StopDynamicMath}
    {\textsubscript{a}}
    {}
    {\LWR@patcherror{erratum}{erratumDelete}}

\xpatchcmd{\erratum}
    {$_d^{\@thefnmark}$}
%     {\StartDynamicMath$_d^{\@thefnmark}$\StopDynamicMath}
    {\textsubscript{d}\textsuperscript{\@thefnmark}}
    {}
    {\LWR@patcherror{erratum}{eDelete}}

\xpatchcmd{\erratum}
    {$_r^{\@thefnmark}$}
%     {\StartDynamicMath$_r^{\@thefnmark}$\StopDynamicMath}
```

```
    {\textsubscript{r}\textsuperscript{\@thefnmark}}
    {}
    {\LWR@patcherror{erratum}{eReplace}}
```

Finish the current page's errata before closing and reloading the list:

```
\preto\PrintErrata{\LWR@orignewpage}
```

No longer defining math macros with the HTML $:

```
\StopDefiningMath
```

File 93 `lwarp-eso-pic.sty`

§ 185 Package eso-pic

(*Emulates or patches code by* ROLF NIEPRASCHK.)

Pkg `eso-pic` **eso-pic** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{eso-pic}

\newcommand*{\LenToUnit}{}
\newcommand{\AtPageUpperLeft}[1]{}
\newcommand{\AtPageLowerLeft}[1]{}
\newcommand{\AtPageCenter}[1]{}
\newcommand{\AtStockLowerLeft}[1]{}
\newcommand{\AtStockUpperLeft}[1]{}
\newcommand{\AtStockCenter}[1]{}
\newcommand{\AtTextUpperLeft}[1]{}
\newcommand{\AtTextLowerLeft}[1]{}
\newcommand{\AtTextCenter}[1]{}
\NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}

\newcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
\NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
\newcommand*{\ClearShipoutPictureBG}{}
\newcommand*{\ClearShipoutPicture}{}
\newcommand*{\ClearShipoutPictureFG}{}
\newcommand{\gridSetup}[6][]{}
```

File 94 `lwarp-everypage.sty`

§ 186 Package everypage

(*Emulates or patches code by* Sergio Callegari.)

Pkg `everypage` **everypage** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{everypage}

\newcommand*{\AddEverypageHook}[1]{}
\newcommand*{\AddThispageHook}[1]{}
```

File 95 `lwarp-everyshi.sty`

§ 187 Package everyshi

(*Emulates or patches code by* Martin Schröder.)

Pkg `everyshi` Emulated.

for HTML output: Discard all options for **lwarp-everyshi**:

```
\LWR@ProvidesPackageDrop{everyshi}

\newcommand*{\EveryShipout}[1]{}
\newcommand*{\AtNextShipout}[1]{}
```

File 96 `lwarp-extramarks.sty`

§ 188 Package extramarks

(*Emulates or patches code by* Piet van Oostrum.)

Pkg `extramarks` **extramarks** is emulated.

for HTML output: Discard all options for **lwarp-extramarks**:

```
\LWR@ProvidesPackageDrop{extramarks}
```

```
\newcommand*{\extramarks}[2]{}
\newcommand*{\firstleftxmark}{}
\newcommand*{\lastleftxmark}{}
\newcommand*{\firstrightxmark}{}
\newcommand*{\lastrightxmark}{}
\newcommand*{\firstxmark}{}
\newcommand*{\lastxmark}{}
\newcommand*{\topxmark}{}
\newcommand*{\topleftxmark}{}
\newcommand*{\firstleftmark}{}
\newcommand*{\lastrightmark}{}
```

File 97 `lwarp-fancybox.sty`

§ 189 Package

fancybox

(*Emulates or patches code by* Timothy Van Zandt.)

Pkg `fancybox`

fancybox is supported with some patches.

framed equation example

fancybox's documentation has an example `FramedEqn` environment which combines math, `\Sbox`, a minipage, and an `\fbox`. This combination requires that the entire environment be enclosed inside a `lateximage`, which is done by adding `\lateximage` at the very start of `FramedEqn`'s beginning code, and `\endlateximage` at the very end of the ending code. Unfortunately, the HTML `alt` attribute is not used here.

```
\newenvironmentFramedEqn
{
\lateximage% NEW
\setlength{\fboxsep}{15pt}
...}{...
\[\fbox{\TheSbox}\]
\endlateximage% NEW
}
```

framing alternatives

`\fbox` works with **fancybox**. Also see **lwarp**'s `\fboxBlock` macro and `fminipage` environment for alternatives to `\fbox` for framing environments.

framed table example

The **fancybox** documentation's example framed table using an `\fbox` containing a `tabular` does not work with **lwarp**, but the `FramedTable` environment does work if `\fbox` is replaced by `\fboxBlock`. This method loses HTML formatting. A better method is to enclose the table's contents inside a `fminipage` environment. The caption may be placed either inside or outside the `fminipage`:

```
\begin{table}
\begin{fminipage}{\linewidth}
\begin{tabular}{lr}
...
\end{tabular}
\end{fminipage}
\end{table}
```

framed verbatim

lwarp does not support the `verbatim` environment inside a span, box, or **fancybox**'s `\Sbox`, but a `verbatim` may be placed inside a `fminipage`. The **fancybox** documentation's example `FramedVerb` may be defined as:

```
\newenvironment{FramedVerb}[1] % width
{
\VerbatimEnvironment
\fminipage{#1}
\beginVerbatim
}{
\endVerbatim
\endfminipage
}
```

framed `\VerbBox`

fancybox's `\VerbBox` may be used inside `\fbox`.

indented alignment

`LVerbatim`, `\LVerbatimInput`, and `\LUseVerbatim` indent with horizontal space which may not line up exactly with what **pdftotext** detects. Some lines may be off slightly in their left edge.

⚠ `\VerbatimFootnotes` ⚠ sectioning or displaymath

If using **fancybox** or **fancyvrb** with `\VerbatimFootnotes`, and using footnotes in a sectioning command or display math, use `\footnotemark` and `\footnotetext`:

```
\subsection[Subsection Name]
    {Subsection Name\protect\footnotemark}
\footnotetext{A footnote with \verb+verbtim+.}
```

and likewise for equations or display math.

At present there is a bug such that paragraph closing tags are not present in footnotes when `\VerbatimFootnotes` are selected. The browser usually compensates.

```
1 \LWR@ProvidesPackagePass{fancybox}
```

After the preamble is loaded, after any patches to `Verbatim`:

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching fancybox.}
```

\VerbatimFootnotes Patched to use the new version.

```
\def\VerbatimFootnotes{%
\let\@footnotetext\V@footnotetext%
\let\LWR@footnotetext\V@footnotetext% lwarp
}
```

\V@@footnotetext Patches in a subset of **lwarp**'s \LWR@footnotetext to the **fancyvrb** version of \V@@footnotetext.

```
\def\V@@footnotetext{%
\LWR@traceinfo{V@footnotetext}%
\global\setbox\LWR@footnotes=\vbox\bgroup%
```

Add to any current footnotes:

```
    \unvbox\LWR@footnotes%
```

Remember the footnote number for \ref:

```
    \protected@edef\@currentlabel{%
        \csname p@footnote\endcsname\@thefnmark%
    }% @currentlabel
```

Use HTML superscripts in the footnote even inside a lateximage:

```
    \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
    \@makefntext{}%

  \bgroup%
  \aftergroup{\V@@@footnotetext}%
  \ignorespaces%
}%

}% AfterEndPreamble

\renewcommand*{\@shadowbox}[1]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px solid black]{shadowbox}{#1}}%
{\InlineClass{shadowbox}{#1}}%
}

```

```
\renewcommand*{\@doublebox}[1]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px double black]{doublebox}{#1}}%
{\InlineClass{doublebox}{#1}}%
}

\renewcommand*{\@ovalbox}[2]{%
\ifbool{FormatWP}%
{\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
{%
    \ifthenelse{\isequivalentto{#1}{\thinlines}}%
    {\InlineClass{ovalbox}{#2}}%
    {\InlineClass{Ovalbox}{#2}}%
}%
}
```

Convert minipages, parboxes, and lists into linear text using the `LWR@nestspan` environment:

```
\let\LWR@origSbox\Sbox

\def\Sbox{\LWR@origSbox\LWR@nestspan}


\let\LWR@origendSbox\endSbox

\def\endSbox{\endLWR@nestspan\LWR@origendSbox}
```

`Beqnarray` is adapted for MathJax or enclosed inside a `lateximage`:

```
\RenewEnviron{Beqnarray}
{\LWR@eqnarrayfactor}

\csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

`\GenericCaption` is enclosed in an HTML block:

```
\renewcommand{\GenericCaption}[1]{%
\LWR@figcaption%
#1%
\endLWR@figcaption%
}
```

`Btrivlist` is enclosed in an HTML block. This is a tabular, and does not use `\item`.

`\trivlist` {⟨*l/c/r*⟩} [⟨*t/c/b*⟩]

```
\RenewDocumentEnvironment{Btrivlist}{m o}
```

```
{%
    \begin{BlockClass}{Btrivlist}%
    \tabular{#1}%
}
{%
    \endtabular%
    \end{BlockClass}%
}
```

`Btrivlist` is also neutralized when used inside a span:

```
\AtBeginEnvironment{LWR@nestspan}{%
\RenewDocumentEnvironment{Btrivlist}{m o}{}{}%
}
```

lwarp's handling of `\item` is patched to accept **fancybox**'s optional arguments:

```
\let\LWRFB@origitemizeitem\LWR@itemizeitem
\let\LWRFB@origdescitem\LWR@descitem

\RenewDocumentCommand{\LWR@itemizeitem}{d()o}{%
    \IfValueTF{#2}{%
        \LWRFB@origitemizeitem[#2]%
    }{%
        \LWRFB@origitemizeitem%
    }%
}

\RenewDocumentCommand{\LWR@descitem}{d()o}{%
    \IfValueTF{#2}{%
        \LWRFB@origdescitem[#2]~%
    }{%
        \LWRFB@origdescitem%
    }%
}


\RenewDocumentCommand{\LWR@nestspanitem}{d()}{%
\if@newlist\else{\LWR@htmltagc{br /}}\fi%
\LWR@origitem%
}
```

The various boxed lists become regular lists:

```
\renewenvironment{Bitemize}[1][]{\begin{itemize}}{\end{itemize}}
\renewenvironment{Benumerate}[1][]{\begin{enumerate}}{\end{enumerate}}
\renewenvironment{Bdescription}[1][]{\begin{description}}{\end{description}}
```

`\boxput` simply prints one then the other argument, side-by-side instead of above and behind:

```
\RenewDocumentCommand{\boxput}{s d() m m}{%
\IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
}
```

Neutralized commands:

```
\RenewDocumentCommand{\fancyput}{s d() m}{}
\RenewDocumentCommand{\thisfancyput}{s d() m}{}

\RenewDocumentCommand{\fancypage}{m m}{}
\RenewDocumentCommand{\thisfancypage}{m m}{}

\def\LandScape#1{}
\def\endLandScape{}
\def\@Landscape#1#2#3{}
\def\endLandscape{}
```

Low-level patches for `UseVerbatim` and friends:

```
\let\LWRFB@UseVerbatim\UseVerbatim
\renewcommand*{\UseVerbatim}[1]{%
\LWR@atbeginverbatim{3}{Verbatim}%
\LWRFB@UseVerbatim{#1}%
\LWR@afterendverbatim{.5}%
}

\let\LWRFB@LUseVerbatim\LUseVerbatim

\renewcommand*{\LUseVerbatim}[1]{%
\LWR@atbeginverbatim{3}{LVerbatim}%
\noindent%
\LWRFB@LUseVerbatim{#1}%
\LWR@afterendverbatim{.5}%
}

\def\@BUseVerbatim[#1]#2{%
\LWR@atbeginverbatim{3}{BVerbatim}%
\LWRFB@UseVerbatim{#2}%
\LWR@afterendverbatim{.5}%
}
```

File 98 `lwarp-fancyheadings.sty`

§ 190 Package fancyheadings

Pkg `fancyheadings` **fancyheadings** is superceded by **fancyhdr**.

for HTML output:

```
\LWR@loadnever{fancyheadings}{fancyhdr}
```

File 99 `lwarp-fancyhdr.sty`

§ 191 Package fancyhdr

(*Emulates or patches code by* Piet van Oostrum.)

Pkg `fancyhdr` **fancyhdr** is nullified.

for HTML output: Discard all options for **lwarp-fancyhdr**:

```
\LWR@ProvidesPackageDrop{fancyhdr}

\newcommand*{\fancyhead}[2][]{}
\newcommand*{\fancyfoot}[2][]{}
\newcommand*{\fancyhf}[2][]{}
\newcommand*{\fancypagestyle}[2]{}
\newcommand*{\lhead}[2][]{}
\newcommand*{\chead}[2][]{}
\newcommand*{\rhead}[2][]{}
\newcommand*{\lfoot}[2][]{}
\newcommand*{\cfoot}[2][]{}
\newcommand*{\rfoot}[2][]{}
\newcommand*{\headrulewidth}{}
\newcommand*{\footrulewidth}{}
\newcommand*{\fancyheadoffset}[2][]{}
\newcommand*{\fancyfootoffset}[2][]{}
\newcommand*{\fancyhfoffset}[2][]{}
\newcommand*{\iffloatpage}[2]{#2}
\newcommand*{\ifftopfloat}[2]{#2}
\newcommand*{\iffbotfloat}[2]{#2}
```

File 100 lwarp-fancyref.sty

§ 192 Package

fancyref

Pkg fancyref **fancyref** is emulated.

for HTML output:

```
\LWR@ProvidesPackagePass{fancyref}
```

To remove the `margin` option, if `\fancyrefhook` is anything other than the `paren` option, then force it to the default instead. (Comparing to the `margin` option was not possible since **lwarp** has revised the meaning of `\mbox` so the comparison failed.)

```
\newcommand*{\LWRfref@parenfancyrefhook}[1]{(#1)}

\ifdefstrequal{\fancyrefhook}{\LWRfref@parenfancyrefhook}
{}{
    \renewcommand*{\fancyrefhook}[1]{#1}%
}
```

Modified to ignore the page number and **varioref**.

```
\renewcommand*{\@f@ref}[4]{%
  \@ifundefined{#1r@#2@#3}{%
    \PackageError{fancyref}{%
      \backslashchar#1ref\space format ``#2''
      undefined\MessageBreak
      for label type ``#3''%
    }{%
      The format ``#2'' was not defined for the label type
      ``#3''\MessageBreak
      and the \backslashchar#1ref\space command.  Perhaps
      you have only misspelled its name.\MessageBreak
      Otherwise you will have to define it with
      \protect\new#1refformat\MessageBreak
      prior to using it.%
    }%
  }{%
    \fancyrefhook{%
      \@nameuse{#1r@#2@#3}%
        {\ref{#3\fancyrefargdelim#4}}%
%         {\pageref{#3\fancyrefargdelim#4}}% original
%         {\@fancyref@page@ref{#3\fancyrefargdelim#4}}% original
        {}% lwarp
        {}% lwarp
    }%
```

```
32   }%
33 }%
```

File 101 `lwarp-fancytabs.sty`

§ 193 Package fancytabs

Pkg `fancytabs` **fancytabs** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fancytabs}

2 \newcommand{\fancytab}[3][RIGHT]{}
3 \newcommand{\fancytabsStyle}[1]{}
4 \newcommand{\fancytabsHeight}[1]{}
5 \newcommand{\fancytabsWidth}[1]{}
6 \newcommand{\fancytabsCount}[1]{}
7 \newcommand{\fancytabsLeftColor}[1]{}
8 \newcommand{\fancytabsRightColor}[1]{}
9 \newcommand{\fancytabsTop}[1]{}
10 \newcommand{\fancytabsTextVPos}[1]{}
11 \newcommand{\fancytabsTextHPos}[1]{}
12 \newcommand{\fancytabsGap}[1]{}
13 \newcommand{\fancytabsFloor}[1]{}
14 \newcommand{\fancytabsRotate}[1]{}
```

File 102 `lwarp-fancyvrb.sty`

§ 194 Package fancyvrb

(*Emulates or patches code by* Timothy Van Zandt.)

Pkg `fancyvrb` **fancyvrb** is supported with some patches.

⚠ `\VerbatimFootnotes` ⚠ sectioning or displaymath

If using **fancybox** or **fancyvrb** with `\VerbatimFootnotes`, and using footnotes in a sectioning command or display math, use `\footnotemark` and `\footnotetext`:

```
\subsection[Subsection Name]
    {Subsection Name\protect\footnotemark}
\footnotetext{A footnote with \verb+verbtim+.}
```

and likewise for equations or display math.

At present there is a bug such that paragraph closing tags are not present in footnotes when \VerbatimFootnotes are selected. The browser usually compensates.

```
\RequirePackage{xcolor}% for \convertcolorspec

\LWR@ProvidesPackagePass{fancyvrb}
```

Initial default patch for fancyvrb:

```
\fvset{frame=none}%
```

After the preamble is loaded, after any patches to Verbatim:

```
\AfterEndPreamble{
\LWR@traceinfo{Patching fancyvrb.}
```

\VerbatimFootnotes Patched to use the new version.

```
\def\VerbatimFootnotes{%
\let\@footnotetext\V@footnotetext%
\let\footnote\V@footnote%
\let\LWR@footnotetext\V@footnotetext% lwarp
}
```

\V@@footnotetext Patches in a subset of **lwarp**'s \LWR@footnotetext to the **fancyvrb** version of \V@@footnotetext.

```
\def\V@@footnotetext{%
\LWR@traceinfo{V@footnotetext}%
\global\setbox\LWR@footnotes=\vbox\bgroup%
```

Add to any current footnotes:

```
    \unvbox\LWR@footnotes%
```

Remember the footnote number for \ref:

```
    \protected@edef\@currentlabel{%
        \csname p@footnote\endcsname\@thefnmark%
    }% @currentlabel
```

Use HTML superscripts in the footnote even inside a lateximage:

```
    \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a lateximage:

```
    \ifthenelse{%
        \boolean{LWR@doingstartpars} \AND%
        \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
    }%
```

```
    {}%
    {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
    \@makefntext{}%
  \bgroup%
  \aftergroup{\V@@@footnotetext}%
  \ignorespaces%
}%

\preto\FVB@Verbatim{\LWR@forcenewpage}
\preto\FVB@LVerbatim{\LWR@forcenewpage}
% \preto\FVB@BVerbatim{\LWR@forcenewpage}% Fails, so done below.
```

Simplified to remove PDF formatting:

```
\def\FV@BeginListFrame@Single{%
  \FV@SingleFrameLine{\z@}%
}
\def\FV@EndListFrame@Single{%
  \FV@SingleFrameLine{\@ne}%
}
\def\FV@BeginListFrame@Lines{%
  \FV@SingleFrameLine{\z@}%
}
\def\FV@EndListFrame@Lines{%
    \FV@SingleFrameLine{\@ne}%
}
\renewcommand*{\FV@SingleFrameSep}{}
```

Adds HTML formatting:

```
\def\FV@BUseVerbatim#1{%
    \LWR@atbeginverbatim[\LWR@FVstyle]{0}{verbatim}%
    \FV@BVerbatimBegin#1\FV@BVerbatimEnd%
    \LWR@afterendverbatim{0}%
}
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
\newcommand*{\LWR@FVstyle}{}
```

The following patches to `Verbatim` are executed at the start and end of the environment, depending on the choice of `frame`. Original code is from the **fancyvrb** package.

```
\newcommand*{\LWR@fvstartnone}{%
\LWR@traceinfo{fvstartnone}%
% \hbox to\z@{
\LWR@atbeginverbatim[\LWR@FVstyle]{0}{verbatim}%
% }%
}

\newcommand*{\LWR@fvendnone}{%
\LWR@traceinfo{fvendnone}%
% \hbox to\z@{
\LWR@afterendverbatim{0}%
% }%
}

\newcommand*{\LWR@fvstartsingle}{%
\LWR@traceinfo{fvstartsingle}%
\LWR@fvstartnone%
\FV@BeginListFrame@Single%
}

\newcommand*{\LWR@fvendsingle}{%
\LWR@traceinfo{fvendsingle}%
\FV@EndListFrame@Single%
\LWR@fvendnone%
}

\newcommand*{\LWR@fvstartline}{%
\LWR@traceinfo{fvstartline}%
\LWR@fvstartnone%
% \setlength{\LWR@templengthone}{\baselineskip}%
\FV@BeginListFrame@Lines%
% \setlength{\baselineskip}{\LWR@templengthone}%
% \setlength{\baselineskip}{5pt}%
}

\newcommand*{\LWR@fvendline}{%
\LWR@traceinfo{fvendline}%
\FV@EndListFrame@Lines%
\LWR@fvendnone%
}
```

The following patches select the start/left/right/end behaviors depending on `frame`. Original code is from the **fancyvrb** package.

```
\newcommand*{\LWR@FVfindbordercolor}{%
```

```
\FancyVerbRuleColor%
\LWR@findcurrenttextcolor%
\color{black}%
}

% border width of \FV@FrameRule
\newcommand*{\LWR@FVborderstyle}[1]{%
padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
\LWR@FVfindbordercolor %
border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt %
solid \LWR@origpound\LWR@tempcolor ; %
}

\def\FV@Frame@none{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}

\FV@Frame@none% default values

\def\FV@Frame@single{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{}}%
\let\FV@BeginListFrame\LWR@fvstartsingle%
\let\FV@LeftListFrame\FV@LeftListFrame@Single%
\let\FV@RightListFrame\FV@RightListFrame@Single%
\let\FV@EndListFrame\LWR@fvendsingle}

\def\FV@Frame@lines{%
\renewcommand*{\LWR@FVstyle}{%
    \LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
}%
\let\FV@BeginListFrame\LWR@fvstartline%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendline}

\def\FV@Frame@topline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}%
\let\FV@BeginListFrame\LWR@fvstartline%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}

\def\FV@Frame@bottomline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\relax%
\let\FV@RightListFrame\relax%
```

```
\let\FV@EndListFrame\LWR@fvendline}

\def\FV@Frame@leftline{%
\renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}%
% To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
\ifx\FancyVerbFillColor\relax%
\let\FV@FrameFillLine\relax%
\else%
\@tempdima\FV@FrameRule\relax%
\multiply\@tempdima-\tw@%
\edef\FV@FrameFillLine{%
{\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}%
\kern-\number\@tempdima sp}}%
\fi%
\let\FV@BeginListFrame\LWR@fvstartnone%
\let\FV@LeftListFrame\FV@LeftListFrame@Single%
\let\FV@RightListFrame\relax%
\let\FV@EndListFrame\LWR@fvendnone}
```

Adds the optional label to the top and bottom edges. Original code is from the **fancyvrb** package.

```
\def\FV@SingleFrameLine#1{%
%   \hbox to\z@{%
%     \kern\leftmargin
    \ifnum#1=\z@\relax
      \let\FV@Label\FV@LabelBegin
    \else
      \let\FV@Label\FV@LabelEnd
    \fi
    \ifx\FV@Label\relax
%       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
    \else
      \ifnum#1=\z@
%         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
        \ifx\FV@LabelPositionTopLine\relax
        \else
        \LWR@FVfindbordercolor
        \LWR@htmltagc{%
            div class="fancyvrblabel" % extra space
                style="color: \LWR@origpound\LWR@tempcolor"%
        }
        \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
        \LWR@htmltagc{/div}
        \fi
      \else
%         \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
        \ifx\FV@LabelPositionBottomLine\relax
        \else
```

```
        \LWR@FVfindbordercolor

        \LWR@htmltagc{%
            div class="fancyvrblabel" % extra space
                style="color: \LWR@origpound\LWR@tempcolor"%
        }
        \LWR@origtextrm{\FV@LabelEnd}
        \LWR@htmltagc{/div}
        \fi
      \fi
    \fi
%     \hss
%     }
}
```

Processes each line, adding optional line numbers. Original code is from the **fancyvrb** package.

```
\def\FV@ListProcessLine#1{%
   \hbox to \hsize{%
%       \kern\leftmargin
      \hbox to \VerbatimHTMLWidth {%
       \ifcsvoid{FV@LeftListNumber}{}{\kern 2.5em}%
        \FV@LeftListNumber%
%       \FV@LeftListFrame
      \FancyVerbFormatLine{#1}%
      \hss%
%       \FV@RightListFrame
      \FV@RightListNumber%
    }%
      \hss% required to avoid underfull hboxes
}
}
```

Env `BVerbatim`

```
\AtBeginEnvironment{BVerbatim}
{%
\LWR@forcenewpage% instead of \preto
\LWR@atbeginverbatim{0}{bverbatim}%
}

\AfterEndEnvironment{BVerbatim}
{%
\LWR@afterendverbatim{0}%
}
```

End of the modifications to make at the end of the preamble:

```
232 } % \AfterEndPreamble
```

File 103 `lwarp-figcaps.sty`

§ 195 Package figcaps

(*Emulates or patches code by* Patrick W. Daly.)

Pkg `figcaps` Emulated.

for HTML output: Discard all options for **lwarp-figcaps**:

```
1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}{}
3 \newcommand*{\figcapsoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}
```

File 104 `lwarp-figsize.sty`

§ 196 Package figsize

(*Emulates or patches code by* Anthony A. Tanbakuchi.)

Pkg `figsize` **figsize** is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{figsize}
```

Emulates a virtual 6×9 inch textsize.

```
2 \newlength{\figwidth}
3 \newlength{\figheight}
4
5 \newcommand{\SetFigLayout}[3][0]{%
6 \setlength{\figheight}{8in}%
7 \setlength{\figheight}{\figheight / #2}%
8 %
```

```
\setlength{\figwidth}{5.5in}%
\setlength{\figwidth}{\figwidth / #3}%
}
```

File 105 `lwarp-fix2col.sty`

§ 197 Package

fix2col

Pkg `fix2col` **fix2col** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fix2col}
```

File 106 `lwarp-fixme.sty`

§ 198 Package

fixme

(*Emulates or patches code by* Didier Verna.)

Pkg `fixme` **fixme** is patched for use by **lwarp**.

⚠ external layouts External layouts (`\fxloadlayouts`) are not supported.

User control is provided for setting the HTML styling of the "faces". The defaults are as follows, and may be changed in the preamble after **fixme** is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

for HTML output:

```
\LWR@ProvidesPackagePass{fixme}
```

Restore **lwarp**'s version of `\@wrindex`, ignoring the **fixme** package's `target` option:

```
\let\@wrindex\LWR@wrindex
```

Float-related macros required by **lwarp**:

```
\newcommand{\ext@fixme}{lox}

\renewcommand{\l@fixme}[2]{\hypertocfloat{1}{fixme}{lox}{#1}{#2}}
```

Other modifications:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}

\renewcommand*\FXLayoutInline[3]{ %
\InlineClass[\FXFaceInlineHTMLStyle]{fixmeinline}%
    {\@fxtextstd{#1}{#2}{#3}}%
}

\def\FXFaceEnvHTMLStyle{font-weight:bold}

\renewcommand*\FXEnvLayoutPlainBegin[2]{%
\BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
\ignorespaces#2 \fxnotename{#1}: \ignorespaces}

\renewcommand*\FXEnvLayoutPlainEnd[2]{\endBlockClass}

\renewcommand*\FXEnvLayoutSignatureBegin[2]{%
\BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
\fxnotename{#1}: \ignorespaces}

\renewcommand*\FXEnvLayoutSignatureEnd[2]{\@fxsignature{#2}\endBlockClass}

\def\FXFaceSignatureHTMLStyle{font-style:italic}

\DeclareRobustCommand*\@fxsignature[1]{%
\ifthenelse{\equal{#1}{}}%
{}%
{ -- {\InlineClass[\FXFaceSignatureHTMLStyle]{fixmesignature}{#1}}}%
}


\def\FXFaceTargetHTMLStyle{font-style:italic}

\renewcommand\FXTargetLayoutPlain[2]{%
    \InlineClass[\FXFaceTargetHTMLStyle]{fixmetarget}{#2}%
}
```

File 107 **lwarp-fixmetodonotes.sty**

§ 199 Package fixmetodonotes

(*Emulates or patches code by* Gioele Barabucci.)

Pkg fixmetodonotes **fixmetodonotes** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{fixmetodonotes}

\renewcommand{\NOTES@addtolist}[2]{%
```

```
    \refstepcounter{NOTES@note}%
%   \phantomsection% REMOVED
    \addcontentsline{notes}{NOTES@note}{%
        \protect\numberline{\theNOTES@note}{{#1}: {#2}}%
    }%
}

\renewcommand{\NOTES@marker}[2]{\fbox{%
    \textcolor{#2}{% WAS \color
        \textbf{#1}}%
    }}

\renewcommand{\NOTES@colorline}[2]{%
  \bgroup%
    \ULon{\LWR@backgroundcolor{#1}{#2}}%
}
```

File 108 `lwarp-flafter.sty`

§ 200 Package flafter

Pkg `flafter` **flafter** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{flafter}
\providecommand\fl@trace[1]{}
```

File 109 `lwarp-float.sty`

§ 201 Package float

(*Emulates or patches code by* Anselm Lingnau.)

Pkg `float` **float** is emulated.

Float styles `boxed` and `ruled` are emulated by CSS and a float class according to style.

The HTML `<figure>` `class` is set to the float type, so CSS may also be used to format the float and its caption, according to float type. Furthermore, an additional class is set to the float style: `plain`, `plaintop`, `boxed`, or `ruled`, so CSS may be used to format by float style as well. Default formatting by CSS is provided for `ruled` and `boxed` styles.

for HTML output:

```
\LWR@ProvidesPackageDrop{float}[2001/11/08]
```

\listof See section 71.2 for the \listof command.

\LWR@floatstyle The default float style:

```
\newcommand*{\LWR@floatstyle}{plain}
```

\newfloat {⟨*1: type*⟩} {⟨*2: placement*⟩} {⟨*3: ext*⟩} [⟨*4: within*⟩]

Emulates the \newfloat command from the **float** package.

"placement" is ignored.

```
\NewDocumentCommand{\newfloat}{m m m o}{%
\IfValueTF{#4}%
{\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}%
{\DeclareFloatingEnvironment[fileext=#3]{#1}}%
```

Remember the float style:

```
\csedef{LWR@floatstyle@#1}{\LWR@floatstyle}%
```

newfloat package automatically creates the \listof command for new floats, but **float** does not, so remove \listof here in case it is manually created later.

```
\cslet{listof#1s}\relax%
\cslet{listof#1es}\relax%
```

Likesize, **newfloat** also creates \l@<type>, but **float** does not, so remove it here:

```
\cslet{l@#1}\relax%
}
```

\floatname {⟨*type*⟩} {⟨*name*⟩}

Sets the text name of the float, such as "Figure".

```
\NewDocumentCommand{\floatname}{m +m}{%
    \SetupFloatingEnvironment{#1}{name=#2}%
}
```

\floatplacement {⟨*type*⟩} {⟨*placement*⟩}

Float placement is ignored.

```
\newcommand*{\floatplacement}[2]{%
    \SetupFloatingEnvironment{#1}{placement=#2}%
}
```

\floatstyle {⟨*style*⟩}

Remember the style for future floats:

```
\newcommand{\floatstyle}[1]{%
    \def\LWR@floatstyle{#1}%
}
```

`\restylefloat` `*` `{⟨type⟩}`

Remember the style for this float:

```
\NewDocumentCommand{\restylefloat}{s m}{%
    \csedef{LWR@floatstyle@#2}{\LWR@floatstyle}%
}
```

File 110 `lwarp-floatflt.sty`

§ 202 Package

floatflt

(*Emulates or patches code by* Mats Dahlgren.)

Pkg `floatflt` Emulated.

for HTML output: Discard all options for **lwarp-floatflt**:

```
\LWR@ProvidesPackageDrop{floatflt}
```

Env `[⟨⟩]` offset `{⟨type⟩}` `{⟨width⟩}` Borrowed from the **lwarp** version of **keyfloat**:

```
\NewDocumentEnvironment{KFLTfloatflt@marginfloat}{O{-1.2ex} m m}
{%
\setlength{\LWR@templengthone}{#3}%
\LWR@BlockClassWP{%
    float:right; %
    width:\LWR@printlength{\LWR@templengthone};  % extra space
    margin:10pt%
}{%
    width:\LWR@printlength{\LWR@templengthone}%
}%
{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}
```

Env `floatingfigure` `[⟨placement⟩]` `{⟨width⟩}`

```
\DeclareDocumentEnvironment{floatingfigure}{o m}
```

```
{\begin{KFLTfloatflt@marginfloat}{figure}{#2}}
{\end{KFLTfloatflt@marginfloat}}
```

Env `floatingtable` [⟨*placement*⟩]

```
\DeclareDocumentEnvironment{floatingtable}{o}
  {\begin{KFLTfloatflt@marginfloat}{table}{1.5in}}
  {\end{KFLTfloatflt@marginfloat}}
```

File 111 `lwarp-floatpag.sty`

§ 203 Package floatpag

(*Emulates or patches code by* Vytas Statulevičius and Sigitas Tolušis.)

Pkg `floatpag` Emulated.

for HTML output: Discard all options for **lwarp-floatpag**:

```
\LWR@ProvidesPackageDrop{floatpag}
```

```
\newcommand*{\floatpagestyle}[1]{}
\newcommand*{\rotfloatpagestyle}[1]{}
\newcommand*{\thisfloatpagestyle}[1]{}
```

File 112 `lwarp-floatrow.sty`

§ 204 Package floatrow

(*Emulates or patches code by* Olga Lapko.)

Pkg `floatrow` **floatrow** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{floatrow}
```

⚠ misplaced alignment alignment tab character &

Use `\StartDefiningTabulars` and `\StopDefiningTabulars` before and after defining macros using `\ttabbox` with a tabular inside. See section 9.9.

⚠ **subfig** package

When combined with the **subfig** package, while inside a `subfloatrow` `\ffigbox` and `\ttabbox` must have the caption in the first of the two of the mandatory arguments.

⚠ `\FBwidth`, `\FBheight`

The emulation of **floatrow** does not support `\FBwidth` or `\FBheight`. These values are pre-set to `.3\linewidth` and `2in`. Possible solutions include:

- Use fixed lengths. **lwarp** will scale the HTML lengths appropriately.
- Use `warpprint` and `warpHTML` environments to select appropriate values for each case.
- Inside a `warpHTML` environment, manually change `\FBwidth` or `\FBheight` before the `\ffigbox` or `\ttabbox`. Use `\FBwidth` or `\FBheight` normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

After everything has loaded, remember whether **subcaption** was loaded. If not, it is assumed that **subfig** is used instead:

```
\newbool{LWR@subcaptionloaded}

\AtBeginDocument{
\@ifpackageloaded{subcaption}
{\booltrue{LWR@subcaptionloaded}}
{\boolfalse{LWR@subcaptionloaded}}
}
```

`\floatbox` `[`⟨*1 preamble*⟩`]` `{`⟨*2 captype*⟩`}` `[`⟨*3 width*⟩`]` `[`⟨*4 height*⟩`]` `[`⟨*5 vert pos*⟩`]` `{`⟨*6 caption*⟩`}` `{`⟨*7 object*⟩`}`

Only parameters for captype, width, caption, and object are used.

`LWR@insubfloatrow` is true if inside a `subfloatrow` environment.

There are two actions, depending on the use of **subcaption** or **subfig**.

```
\NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
\ifbool{LWR@subcaptionloaded}%
{% subcaption
```

For **subcaption**:

```
    \ifbool{LWR@insubfloatrow}%
    {% subcaption in a subfloatrow
```

`subfigure` and `subtable` environments take width as an argument.

```
        \IfValueTF{#3}%
        {\@nameuse{sub#2}{#3}}%
        {\@nameuse{sub#2}{\linewidth}}%
    }% subcaption in a subfloatrow
    {% subcaption not in subfloatrow
```

`figure` and `table` environments do not take a width argument.

```
        \@nameuse{#2}%
    }% subcaption not in subfloatrow
    #6
```

```
22
23     #7
```

End the environments:

```
24     \ifbool{LWR@insubfloatrow}%
25     {\@nameuse{endsub#2}}%
26     {\@nameuse{end#2}}%
27 }% subcaption
28 {% assume subfig
```

For **subfig**:

```
29 \ifbool{LWR@insubfloatrow}%
30 {% subfig in a subfloatrow
```

`\subfloat` is a macro, not an environment.

Package **subfig**'s `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument #6 contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```
31     \begingroup
32     \let\caption\@firstofone
33     \subfloat[#6]{#7}
34     \endgroup
35 }% subfig in a subfloatrow
36 {% subfig package, but not a subfig
```

`figure` and `table` are environments:

```
37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }
```

Not used:

```
45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][]{}
54 \newcommand{\thisfloatsetup}[1]{}
55 \newcommand{\clearfloatsetup}[1]{}
```

```
\newcommand*{\killfloatstyle}{}
```

`\newfloatcommand` {⟨*1 command*⟩} {⟨*2 captype*⟩} [⟨*3 preamble*⟩] [⟨*4 default width*⟩]

Preamble and default width are ignored.

```
\NewDocumentCommand{\newfloatcommand}{m m o o}{%
\@namedef{#1}{
\floatbox{#2}
}
}
```

`\renewfloatcommand` {⟨*1 command*⟩} {⟨*2 captype*⟩} [⟨*3 preamble*⟩] [⟨*4 default width*⟩]

Preamble and default width are ignored.

```
\NewDocumentCommand{\renewfloatcommand}{m m o o}{%
\@namedef{#1}{%
\floatbox{#2}
}
}
```

`\ffigbox` [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{ffigbox}{figure}[\nocapbeside][]
```

`\ttabbox` [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{ttabbox}{table}[\captop][\FBwidth]
```

`\fcapside` [⟨*width*⟩] [⟨*height*⟩] [⟨*vposn*⟩] {⟨*caption commands*⟩} {⟨*contents*⟩}

```
\newfloatcommand{fcapside}{figure}[\capbeside][]
```

Env `floatrow` [⟨*numfloats*⟩]

The row of floats is placed into a `<div>` of class `floatrow`.

```
\newenvironment*{floatrow}[1][2]
{
\LWR@forcenewpage
\BlockClass{floatrow}
```

While inside the floatrow, divide the `\linewidth` by the number of floats.

```
\booltrue{LWR@infloatrow}
\setlength{\linewidth}{6in/#1}
}
{
\boolfalse{LWR@infloatrow}
\endBlockClass
```

```
}
```

Keys for \DeclareNewFloatType:

```
\newcommand*{\LWR@frowkeyplacement}{}
\newcommand*{\LWR@frowkeyname}{}
\newcommand*{\LWR@frowkeyfileext}{}
\newcommand*{\LWR@frowkeywithin}{}
\newcommand*{\LWR@frowkeycapstyle}{}

\define@key{frowkeys}{placement}{}%
\define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
\define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
\define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
\define@key{frowkeys}{relatedcapstyle}{}%
```

\DeclareNewFloatType {⟨*type*⟩} {⟨*options*⟩}

Use \listof{type}{Title} to print a list of the floats.

```
\newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
\renewcommand*{\LWR@frowkeyplacement}{}%
\renewcommand*{\LWR@frowkeyname}{}%
\renewcommand*{\LWR@frowkeyfileext}{}%
\renewcommand*{\LWR@frowkeywithin}{}%
\renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```
\LWR@traceinfo{about to setkeys frowkeys}%
\setkeys{frowkeys}{#2}%
\LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```
\ifthenelse{\equal{\LWR@frowkeywithin}{}}%
{%
    \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\ %
        \LWR@frowkeyfileext}%
    \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}%
}%
{%
    \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\ %
        \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
    \newfloat{#1}{\LWR@frowkeyplacement}%
    {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
    \LWR@traceinfo{finished newfloat #1}%
}%
```

Rename the float if a name was given:

```
\ifthenelse{\equal{\LWR@frowkeyname}{}}%
{}%
{\floatname{#1}{\LWR@frowkeyname}}%
}
```

Not used:

```
\newcommand{\buildFBBOX}[2]{}
\newcommand*{\CenterFloatBoxes}{}
\newcommand*{\TopFloatBoxes}{}
\newcommand*{\BottomFloatBoxes}{}
\newcommand*{\PlainFloatBoxes}{}

\newcommand{\capsubrowsettings}{}

\NewDocumentCommand{\RawFloats}{o o}{}
```

`\RawCaption` `{`⟨*text*⟩`}`

To be used inside a minipage or parbox.

```
\newcommand{\RawCaption}[1]{#1}
```

`\floatfoot` `{`⟨*text*⟩`}`

Places additional text inside a float, inside a css `<div>` of class `floatfoot`.

```
\NewDocumentCommand{\floatfoot}{s +m}{%
\begin{BlockClass}{floatfoot}
#2
\end{BlockClass}
}
```

Used to compute `\linewidth`.

```
\newbool{LWR@insubfloatrow}
\boolfalse{LWR@insubfloatrow}
```

Env `subfloatrow` `[`⟨*num_floats*⟩`]`

```
\newenvironment*{subfloatrow}[1][2]
{
```

The row of floats is placed into a `<div>` of class `floatrow`:

```
\LWR@forcenewpage
\BlockClass{floatrow}
```

While inside the floatrow, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
\begingroup
\booltrue{LWR@insubfloatrow}
}
{
\endgroup
\endBlockClass
\boolfalse{LWR@insubfloatrow}
}
```

File 113 `lwarp-fltrace.sty`

§ 205 Package fltrace

Pkg `fltrace` **fltrace** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fltrace}

\def\tracefloats{}
\def\tracefloatsoff{}
\def\tracefloatvals{}
```

File 114 `lwarp-flushend.sty`

§ 206 Package flushend

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `flushend` Emulated.

for HTML output: Discard all options for **lwarp-flushend**:

```
\LWR@ProvidesPackageDrop{flushend}

\newcommand*{\flushend}{}
\newcommand*{\raggedend}{}
\newcommand*{\flushcolsend}{}
\newcommand*{\raggedcolsend}{}
\newcommand*{\atColsBreak}[1]{}
\newcommand*{\atColsEnd}[1]{}
\newcommand*{\showcolsendrule}{}
```

File 115 `lwarp-fnbreak.sty`

§ 207 Package fnbreak

Pkg `fnbreak` **fnbreak** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fnbreak}

\newcommand*{\fnbreakverbose}{}
\newcommand*{\fnbreaknonverbose}{}
\newcommand*{\fnbreaklabel}{}
\newcommand*{\fnbreaknolabel}{}
```

File 116 `lwarp-fncychap.sty`

§ 208 Package fncychap

(*Emulates or patches code by* Ulf A. Lindgren.)

Pkg `fncychap` **fncychap** is emulated.

for HTML output: Discard all options for **lwarp-fncychap**:

```
\LWR@ProvidesPackageDrop{fncychap}

\def\mghrulefill#1{}
\def\ChNameLowerCase{}
\def\ChNameUpperCase{}
\def\ChNameAsIs{}
\def\ChTitleLowerCase{}
\def\ChTitleUpperCase{}
\def\ChTitleAsIs{}
\newcommand{\ChRuleWidth}[1]{}
\newcommand{\ChNameVar}[1]{}
\newcommand{\ChNumVar}[1]{}
\newcommand{\ChTitleVar}[1]{}
\newcommand{\TheAlphaChapter}{}
\newcommand{\DOCH}{}
\newcommand{\DOTI}[1]{}
\newcommand{\DOTIS}[1]{}
\newlength{\mylen}
\newlength{\myhi}
```

```
\newlength{\px}
\newlength{\py}
\newlength{\pyy}
\newlength{\pxx}
\newlength{\RW}
\newcommand{\FmN}[1]{#1}
\newcommand{\FmTi}[1]{#1}
```

File 117 `lwarp-fnlineno.sty`

§ 209 Package fnlineno

Pkg `fnlineno` **fnlineno** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fnlineno}
```

File 118 `lwarp-fnpos.sty`

§ 210 Package fnpos

(*Emulates or patches code by* Hiroshi Nakashima.)

Pkg `fnpos` **fnpos** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{fnpos}

\newcommand*{\makeFNbottom}{}
\newcommand*{\makeFNmid}{}
\newcommand*{\makeFNbelow}{}
\newcommand*{\makeFNabove}{}
```

File 119 `lwarp-fontenc.sty`

§ 211 Package fontenc

Pkg `fontenc` If using pdfLaTeX, **lwarp** used to require **fontenc** be loaded before **lwarp**, but now **lwarp** itself loads `\fontenc` with T1 encoding, which **lwarp** requires. **fontenc** is now allowed to be loaded with another encoding after **lwarp**.

lwarp-fontenc is no longer necessary, but is still provided to overwrite older versions.

for HTML output:

```
\LWR@ProvidesPackagePass{fontenc}
```

File 120 `lwarp-fontspec.sty`

§ 212 Package fontspec

Pkg `fontspec` Error if **fontspec** is loaded after **lwarp**.

Discard all options for **lwarp-fontspec**:

for HTML output:

```
\LWR@ProvidesPackageDrop{fontspec}[2017/11/09]
\LWR@loadbefore{fontspec}
```

File 121 `lwarp-footmisc.sty`

§ 213 Package footmisc

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `footmisc` **footmisc** is emulated.

lwarp incidentally happens to emulate the `stable` option.

```
\LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
\newcommand{\footnotelayout}{}
\newcommand{\setfnsymbol}[1]{}
\NewDocumentCommand{\DefineFNsymbols}{s m o m}{}

\newdimen\footnotemargin
\footnotemargin1.8em\relax

\newcommand*\hangfootparskip{0.5\baselineskip}
\newcommand*\hangfootparindent{0em}%

\let\pagefootnoterule\footnoterule
\let\mpfootnoterule\footnoterule
\def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}

\providecommand*{\multiplefootnotemarker}{3sp}
\providecommand*{\multfootsep}{,}
```

Using **cleveref**:

```
\providecommand*{\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
\newcommand\mpfootnotemark{%
  \@ifnextchar[%
    \@xmpfootnotemark%
    {%
      \stepcounter\@mpfn%
      \protected@xdef\@thefnmark{\thempfn}%
      \@footnotemark%
    }%
}
\def\@xmpfootnotemark[#1]{%
  \begingroup%
    \csname c@\@mpfn\endcsname #1\relax%
    \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup%
  \@footnotemark%
}
```

File 122 `lwarp-footnote.sty`

§ 214 Package

footnote

(*Emulates or patches code by* Mark Wooding.)

Pkg `footnote` **footnote** is used with minor patches.

for HTML output:

```
\LWR@ProvidesPackagePass{footnote}
```

Removed print-version formatting:

```
\def\fn@startnote{%
%   \@parboxrestore%
  \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
%   \color@begingroup% *** conflicts with lwarp
}

% \let\fn@endnote\color@endgroup% *** conflicts with lwarp
\def\fn@endnote{%
\LWR@htmltagc{/\LWR@tagregularparagraph}%
\LWR@orignewline%
}
```

Removed print-version formatting:

```
\def\fn@startfntext{%
  \setbox\z@\vbox\bgroup%
    \fn@startnote%
    \fn@prefntext%
    \ignorespaces%
}
```

Removed print-version formatting, added closing paragraph tag:

```
\def\fn@endfntext{%
    \LWR@htmltagc{/\LWR@tagregularparagraph}%
    \LWR@orignewline%
    \fn@postfntext%
  \egroup%
  \begingroup%
    \let\@makefntext\@empty%
    \let\@finalstrut\@gobble%
    \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
    \@footnotetext{\unvbox\z@}%
  \endgroup%
}
```

These have been redefined, so re-`\let` them again:

```
\let\endfootnote\fn@endfntext
\let\endfootnotetext\endfootnote
```

File 123 `lwarp-footnotehyper.sty`

§ 215 Package

footnotehyper

Pkg `footnotehyper`

footnotehyper is a **hyperref**-safe version of **footnote**. For **lwarp**, **footnotehyper** is emulated.

for HTML output:

Discard all options for **lwarp-footnotehyper**:

```
\RequirePackage{footnote}
\LWR@ProvidesPackageDrop{footnotehyper}
```

File 124 `lwarp-footnpag.sty`

§ 216 Package footnpag

Pkg `footnpag` **footnpag** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{footnpag}
```

File 125 `lwarp-forest.sty`

§ 217 Package forest

(*Emulates or patches code by* Sašo Živanović.)

Pkg `forest` **forest** is patched for use by **lwarp**.

⚠ `\Forest*` The starred version of the macro `\Forest*` is not supported. **lwarp** encases each `lateximage` in an environment, so the global results of the starred `\Forest*` are lost.

for HTML output:

```
\LWR@ProvidesPackagePass{forest}

\BeforeBeginEnvironment{forest}{\begin{lateximage}[forest]}

\AfterEndEnvironment{forest}{\end{lateximage}}

\RenewDocumentCommand{\Forest}{s D(){} m}{%
  \forest@config{#2}%
  \IfBooleanTF{#1}{%
        \PackageError{lwarp-forest}%
        {Starred \Forest is not supported}%
        {Lwarp uses an environment for images, but \Forest* cannot work in an environment.}%
        \let\forest@next\forest@env%
    }{\let\forest@next\forest@group@env}%
    \begin{lateximage}[forest]%     lwarp
  \forest@next{#3}%
    \end{lateximage}%               lwarp
}
```

File 126 `lwarp-framed.sty`

§ 218 Package framed

(*Emulates or patches code by* Donald Arseneau.)

Pkg `framed` **framed** is supported and patched by **lwarp**.

for HTML output: Accept all options for **lwarp-framed**:

```
\LWR@ProvidesPackagePass{framed}
\RequirePackage{xcolor}% for \convertcolorspec


\renewenvironment{framed}{%
\LWR@forcenewpage
\BlockClass{framed}%
}
{\endBlockClass}

\renewenvironment{oframed}{%
\LWR@forcenewpage
\BlockClass{framed}%
}
{\endBlockClass}


\renewenvironment{shaded}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \LWR@origpound\LWR@tempcolor]{shaded}%
}
{\endBlockClass}

\renewenvironment{shaded*}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \LWR@origpound\LWR@tempcolor]{shaded}%
}
{\endBlockClass}


\renewenvironment{leftbar}{%
\LWR@forcenewpage
    \BlockClass{framedleftbar}
  \def\FrameCommand{}%
```

```
  \MakeFramed {}
}%
 {\endMakeFramed\endBlockClass}


\renewenvironment{snugshade}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \LWR@origpound\LWR@tempcolor]{snugframed}%
}
{\endBlockClass}

\renewenvironment{snugshade*}{%
\convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\BlockClass[background: \LWR@origpound\LWR@tempcolor]{snugframed}%
}
{\endBlockClass}

\let\oframed\framed
\let\endoframed\endframed


\RenewEnviron{titled-frame}[1]{%
\CustomFBox{#1}{}{0pt}{0pt}{0pt}{0pt}{\BODY}
}
```

\CustomFBox {⟨*toptitle*⟩} {⟨*bottitle*⟩} {⟨*thicknesstop*⟩} {⟨*bottom*⟩} {⟨*left*⟩} {⟨*right*⟩} {⟨*text contents*⟩}

```
\renewcommand{\CustomFBox}[7]{%
\convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
\LWR@forcenewpage
\begin{BlockClass}[border: 3px solid \LWR@origpound\LWR@tempcolor]{framed}%
\ifthenelse{\isempty{#1}}{}{% not empty
    \begin{BlockClass}[background: \LWR@origpound\LWR@tempcolor]{framedtitle}%
    \textcolor{TFTitleColor}{\textbf{#1}}%
    \end{BlockClass}
}% not empty

#7

\ifthenelse{\isempty{#2}}{}{% not empty
    \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
    \begin{BlockClass}[background: \LWR@origpound\LWR@tempcolor]{framedtitle}%
    \textcolor{TFTitleColor}{\textbf{#2}}%
    \end{BlockClass}
}% not empty
\end{BlockClass}
```

```
}
```

`\TitleBarFrame` [⟨*marker*⟩] {⟨*title*⟩} {⟨*contents*⟩}

```
\renewcommand\TitleBarFrame[3][]{
\CustomFBox
    {#2}{}%
    \fboxrule\fboxrule\fboxrule\fboxrule
    {#3}%
}

\renewcommand{\TF@Title}[1]{#1}
```

`MakeFramed` {⟨*settings*⟩}

```
\let\MakeFramed\relax
\let\endMakeFramed\relax

\NewEnviron{MakeFramed}[1]{%
\FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
}
```

`\fb@put@frame` {⟨*frame cmd no split*⟩} {⟨*frame cmd split*⟩}

```
\renewcommand*{\fb@put@frame}[2]{%
\relax%
\@tempboxa%
}
```

File 127 `lwarp-ftnright.sty`

§ 219 Package ftnright

Pkg `ftnright` **ftnright** is ignored.

for HTML output: Discard all options for **lwarp-ftnright**:

```
\LWR@ProvidesPackageDrop{ftnright}
```

File 128 `lwarp-fullminipage.sty`

§ 220 Package fullminipage

Pkg `fullminipage` **fullminipage** is nullified.

for HTML output:

```
\LWR@ProvidesPackageDrop{fullminipage}[2014/07/06]
\newenvironment{fullminipage}[1][]{}{}
```

File 129 `lwarp-fullpage.sty`

§ 221 Package fullpage

Pkg `fullpage` **fullpage** is ignored.

for HTML output: Discard all options for **lwarp-fullpage**:

```
\LWR@ProvidesPackageDrop{fullpage}[1994/06/01]
```

File 130 `lwarp-fullwidth.sty`

§ 222 Package fullwidth

(*Emulates or patches code by* Marco Daniel.)

Pkg `fullwidth` **fullwidth** is emulated.

A minipage is used, of no HTML width.

for HTML output:

```
\LWR@ProvidesPackageDrop{fullwidth}
\newenvironment*{fullwidth}[1][]{%
\minipagefullwidth%
\minipage{\linewidth}%
}
{%
\endminipage%
}
```

File 131 `lwarp-fwlw.sty`

§ 223 Package fwlw

Pkg `fwlw` **fwlw** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{fwlw}
```

```
\newbox\FirstWordBox      \global\setbox\FirstWordBox\hbox{}
\newbox\NextWordBox       \global\setbox\NextWordBox\hbox{}
\newbox\LastWordBox       \global\setbox\LastWordBox\hbox{}
\def\ps@fwlwhead{}
\def\ps@NextWordFoot{}
```

File 132 `lwarp-geometry.sty`

§ 224 Package geometry

(*Emulates or patches code by* Hideo Umeki.)

Pkg geometry **geometry** is preloaded by **lwarp**, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for **lwarp-geometry**:

```
\LWR@ProvidesPackageDrop{geometry}

\renewcommand*{\geometry}[1]{}
\renewcommand*{\newgeometry}[1]{}
\renewcommand*{\restoregeometry}{}
\renewcommand*{\savegeometry}[1]{}
\renewcommand*{\loadgeometry}[1]{}
```

File 133 `lwarp-glossaries.sty`

§ 225 Package glossaries

(*Emulates or patches code by* Nicola L.C. Talbot.)

Pkg glossaries

processing glossaries

Opt GlossaryCmd

Default: makeglossaries

Opt lwarpmk printglossary

Opt lwarpmk htmlglossary

lwarpmk has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary`, which process the glossaries created by the **glossaries** package using that package's ***makeglossaries*** command.

The shell command to execute is set by the **lwarp** option `GlossaryCmd`, which defaults to ***makeglossaries***. The print or HTML glossary filename is appended to this command.

⚠ ***makeglossaries*** not found

In some situations it may be required to modify the default command, such as to add the ***perl*** command in front:

```
\usepackage[
   GlossaryCmd={perl makeglossaries},
] {lwarp}
```

xindy language

To set the language to use for processing glossaries with **xindy**:

```
\usepackage[
   GlossaryCmd={makeglossaries -L english},
] {lwarp}
```

Other options for ***makeglossaries*** may be set as well.

placement and TOC options

The glossaries may be placed in a numbered or unnumbered section, given a TOC entry, and placed inline or on their own HTML page:

Numbered section, on its own HTML page:

```
\usepackage[xindy,toc,numberedsection=nolabel]{glossaries}
...
\printglossaries
```

Unnumbered section, inline with the current HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\printglossaries
```

Unnumbered section, on its own HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\ForceHTMLPage
\printglossaries
```

⚠ glossary style

The default `style=item` option for **glossaries** conflicts with **lwarp**, so the style is forced to `index` instead.

⚠ number list

The page number list in the printed form would become `\namerefs` in HTML, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

print/HTML versions

The print and HTML versions of the glossary differ in their internal page numbers. Separate commands for generating print and HTML glossaries are used, even though the page number is currently ignored.

for HTML output:

```
\PassOptionsToPackage{xindy}{glossaries}
\LWR@ProvidesPackagePass{glossaries}
\setupglossaries{nonumberlist}
\setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```
\renewcommand*{\@p@glossarysection}[2]{%
  \glsclearpage
```

```
  \phantomsection
  \ifdefempty\@@glossarysecstar
  {%
    \csname\@@glossarysec\endcsname{#2}%
  }%
  {%
```

In the original, the TOC entry was made before the section, thus linking to the phantomsection in the printed version, but for HTML this caused the link to point to the page before the glossaries, which could be a different HTML file. Here, the TOC entry is made after the section is created:

```
      \csname\@@glossarysec\endcsname*{#2}%
    \@gls@toc{#1}{\@@glossarysec}% Moved after the previous line.
  }%
  \@@glossaryseclabel
}
```

lwarp's sectioning commands cannot handle robust macros when splitting HTML into named filenames. **glossaries** uses `\translate` in sectioning names, and `\translate` is robust and cannot be expanded. The following pre-expands the translations at this moment, making use of `\translatelet`.

```
\newcommand*{\LWR@comp@glossaryname}{\translate{Glossary}}

\ifdefstrequal{\glossaryname}{\LWR@comp@glossaryname}{
    \translatelet\LWR@translatetemp{Glossary}
    \edef\glossaryname{\LWR@translatetemp}
}{}

\newcommand*{\LWR@comp@acronymname}{\translate{Acronym}}

\ifdefstrequal{\acronymname}{\LWR@comp@acronymname}{
    \translatelet\LWR@translatetemp{Acronym}
    \edef\acronymname{\LWR@translatetemp}
}{}

\newcommand*{\LWR@comp@glssymbolsgroupname}{\translate{Symbols (glossaries)}}

\ifdefstrequal{\glssymbolsgroupname}{\LWR@comp@glssymbolsgroupname}{
    \translatelet\LWR@translatetemp{Symbols (glossaries)}
    \edef\glssymbolsgroupname{\LWR@translatetemp}
}{}

\newcommand*{\LWR@comp@glsnumbersgroupname}{\translate{Numbers (glossaries)}}

\ifdefstrequal{\glsnumbersgroupname}{\LWR@comp@glsnumbersgroupname}{
    \translatelet\LWR@translatetemp{Numbers (glossaries)}
    \edef\glsnumbersgroupname{\LWR@translatetemp}
```

```
}{}
```

File 134 `lwarp-glossary.sty`

§ 226 Package glossary

Pkg glossary **glossary** is superceded by **glossaries**.

for HTML output:

```
\LWR@loadnever{glossary}{glossaries}
```

File 135 `lwarp-graphics.sty`

§ 227 Package graphics

(*Emulates or patches code by* D. P. Carlisle.)

Pkg graphics **graphics** is emulated.

for HTML output:

```
\LWR@ProvidesPackagePass{graphics}
```

§ 227.1 Graphics extensions

`\DeclareGraphicsExtensions` {⟨*list*⟩}

`\AtBeginDocument` allow SVG files instead of PDF:

```
\AtBeginDocument{
\DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
\DeclareGraphicsRule{.svg}{svg}{.svg}{}
\DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
}
```

Inside a `lateximage`, allow PDF instead of SVG:

```
\appto\LWR@restoreorigformatting{%
\DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
}
```

§ 227.2 Length conversions and graphics options

⚠ whitespace

A scaled image in LaTeX by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

Used to store the user's selected dimensions and HTML class.

The class defaults to "inlineimage" unless changed by a `class=xyx` option.

```
\newlength{\LWR@igwidth}
\newlength{\LWR@igheight}
\newcommand*{\LWR@igwidthstyle}{}
\newcommand*{\LWR@igheightstyle}{}
\newcommand*{\LWR@igorigin}{}
\newcommand*{\LWR@igangle}{}
\newcommand*{\LWR@igxscale}{1}
\newcommand*{\LWR@igyscale}{1}
\newcommand*{\LWR@igclass}{inlineimage}
```

Set the actions of each of the key/value combinations for `\includegraphics`. Many are ignored.

If an optional width was given, set an HTML style:

```
\define@key{igraph}{width}{%
\setlength{\LWR@igwidth}{#1}%
\ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}%
{%
```

Default to use the converted fixed length given:

```
    \renewcommand*{\LWR@igwidthstyle}{width:\LWR@printlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
    \IfEndWith{#1}{ex}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
    {}% not ex
    \IfEndWith{#1}{em}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
    {}% not em
    \IfEndWith{#1}{\%}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
    {}% not percent
    \IfEndWith{#1}{px}%
    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
    {}% not px
}{}% end of length > 0pt
```

```
}
```

If an optional height was given, set an HTML style:

```
\define@key{igraph}{height}{%
\setlength{\LWR@igheight}{#1}%
\ifthenelse{\lengthtest{\LWR@igheight > 0pt}}%
{%
```

Default to use the converted fixed length given:

```
    \renewcommand*{\LWR@igheightstyle}{%
    height:\LWR@printlength{\LWR@igheight} % extra space
    }%
```

If ex or em dimensions were given, use those instead:

```
    \IfEndWith{#1}{ex}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
    {}% not ex
    \IfEndWith{#1}{em}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
    {}% not em
    \IfEndWith{#1}{\%}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
    {}% not percent
    \IfEndWith{#1}{px}%
    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
    {}% not px
}{}% end of length > 0pt
}
```

Handle origin key:

```
\define@key{igraph}{origin}[c]{%
\renewcommand*{\LWR@igorigin}{#1}%
}
```

Handle angle key:

```
\define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}
```

Handle class key:

```
\define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}

```

It appears that **graphicx** does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```
\define@key{igraph}{scale}{%
\renewcommand*{\LWR@igxscale}{#1}%
\renewcommand*{\LWR@igyscale}{#1}}
```

Numerous ignored keys:

```
\define@key{igraph}{bb}{}
\define@key{igraph}{bbllx}{}
\define@key{igraph}{bblly}{}
\define@key{igraph}{bburx}{}
\define@key{igraph}{bbury}{}
\define@key{igraph}{natwidth}{}
\define@key{igraph}{natheight}{}
\define@key{igraph}{hiresbb}[true]{}
\define@key{igraph}{viewport}{}
\define@key{igraph}{trim}{}
\define@key{igraph}{totalheight}{}
\define@key{igraph}{keepaspectratio}[true]{}
\define@key{igraph}{clip}[true]{}
\define@key{igraph}{draft}[true]{}
\define@key{igraph}{type}{}
\define@key{igraph}{ext}{}
\define@key{igraph}{read}{}
\define@key{igraph}{command}{}
```

New in v1.1a:

```
\define@key{igraph}{quite}{}
\define@key{igraph}{page}{}
\define@key{igraph}{pagebox}{}
\define@key{igraph}{interpolate}[true]{}
```

New in v1.1b:

```
\define@key{igraph}{decodearray}{}
```

§ 227.3 Printing HTML styles

`\LWR@rotstyle` `{`⟨*prefix*⟩`}` `{`⟨*degrees*⟩`}`

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```
\newcommand*{\LWR@rotstyle}[2]{%
 #1transform:rotate(-#2deg);
}
```

`\LWR@scalestyle` {⟨*prefix*⟩} {⟨*xscale*⟩} {⟨*yscale*⟩}

Prints the scale style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```
\newcommand*{\LWR@scalestyle}[3]{%
 #1transform:scale(#2,#3);
}
```

§ 227.4 \includegraphics

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
\newbool{LWR@infloatrow}
\boolfalse{LWR@infloatrow}
```

`\LWR@opacity` For HTML, used only for `\includegraphics`.

`\LWR@opacity` may be set by the **transparent** package.

```
\def\LWR@opacity{1}
```

`\LWR@imagesizebox` Used to determine the actual image size if needed.

```
\newsavebox{\LWR@imagesizebox}
```

`\LWR@HTML@Gin@setfile` {⟨*w*⟩} {⟨*h*⟩} {⟨*filename*⟩} Sets the parsed filename for HTMLoutput.

```
\newcommand*{\LWR@HTML@Gin@setfile}[3]{%
    \xdef\LWR@parsedfilename{#3}%
}
```

Key `Gin` `class` CSS class for the image.

Define the new class key for the print-mode version of `\includegraphics`, which is enabled inside a `lateximage`.

```
\AtBeginDocument{
\define@key{Gin}{class}{}
}
```

\LWR@replaceEPSSVG Usually, references to EPS files become SVG files, but if the **epstopdf** package is being used, it automatically converts EPS to PDF, and the following must NOT be done.

```
\AtBeginDocument{
\@ifpackageloaded{epstopdf}
{
    \newcommand*{\LWR@replaceEPSSVG}{}
}{%
    \newcommand*{\LWR@replaceEPSSVG}{%
        \StrSubstitute{\LWR@tempone}{.eps}{.svg}[\LWR@tempone]%
        \StrSubstitute{\LWR@tempone}{.EPS}{.SVG}[\LWR@tempone]%
    }
}%
}
```

\LWR@includegraphicsb * [⟨*2: options*⟩] [⟨*3: options*⟩] {⟨*4: filename*⟩}

graphics syntax is \includegraphics * [⟨*llx,lly*⟩] [⟨*urx,ury*⟩] {⟨*file*⟩}

graphicx syntax is \includegraphics [⟨*key values*⟩] {⟨*file*⟩}

If #3 is empty, only one optional argument was given, thus **graphicx** syntax.

```
\NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
{%
\LWR@traceinfo{LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
\LWR@origtilde \LWR@orignewline%
```

Temporarily compute \linewidth, \textwidth, \textheight arguments with a 6x9 inch size until the next \endgroup.

```
\begingroup%
\ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
{%
    \ifbool{LWR@infloatrow}%
    {}
    {% not in a minipage or a floatrow:
        \setlength{\linewidth}{6in}%
        \setlength{\textwidth}{6in}%
        \setlength{\textheight}{9in}%
    }%
}{}%
```

Fully expand and detokenize the filename, changing the file extension to `.svg` if necessary.

```
\begingroup%
\LetLtxMacro\Gin@setfile\LWR@HTML@Gin@setfile%
\edef\LWR@tempone{#4}%
\StrSubstitute{\LWR@tempone}{.pdf}{.svg}[\LWR@tempone]%
\StrSubstitute{\LWR@tempone}{.PDF}{.SVG}[\LWR@tempone]%
\LWR@replaceEPSSVG%
\xdef\LWR@parsedfilename{\LWR@tempone}%
\Ginclude@graphics{\detokenize\expandafter{\LWR@tempone}}%
\endgroup%
\filename@parse{\LWR@parsedfilename}%
\LWR@traceinfo{LWR@parsedfilename is \LWR@parsedfilename}%
% \LWR@sanitize{\LWR@parsedfilename}%
```

For correct em sizing during the width and height conversions:

```
\large%
```

Reset some defaults, possibly will be changed below if options were given:

```
\setlength{\LWR@igwidth}{0pt}%
\setlength{\LWR@igheight}{0pt}%
\renewcommand*{\LWR@igwidthstyle}{}%
\renewcommand*{\LWR@igheightstyle}{}%
\renewcommand*{\LWR@igorigin}{}%
\renewcommand*{\LWR@igangle}{}%
\renewcommand*{\LWR@igxscale}{1}%
\renewcommand*{\LWR@igyscale}{1}%
\renewcommand*{\LWR@igclass}{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus **graphicx** syntax:

```
\IfValueF{#3}{%
    \IfValueTF{#2}%
        {\setkeys{igraph}{#2}}%
        {\setkeys{igraph}{}}%
}%
```

If formatting for a word processor, find and set the actual image size, without rotation, using PDF instead of SVG to find the original bounding box:

```
\ifbool{FormatWP}{%
    \begingroup%
    \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
    \define@key{Gin}{angle}{}%
    \IfBooleanTF{#1}%
    {% starred
```

```
        \IfValueTF{#3}%
        {%
            \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*[#2][#3]{#4}}%
        }%
        {%
            \IfValueTF{#2}%
            {%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*[#2]{#4}}%
            }{%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics*{#4}}%
            }%
        }%
    }% starred
    {% not starred
        \IfValueTF{#3}%
        {%
            \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics[#2][#3]{#4}}%
        }%
        {%
            \IfValueTF{#2}%
            {%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics[#2]{#4}}%
            }{%
                \global\sbox{\LWR@imagesizebox}{\LWR@origincludegraphics{#4}}%
            }%
        }%
    }% not starred
    \endgroup%
    \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
    \global\renewcommand*{\LWR@igwidthstyle}{width:\LWR@printlength{\LWR@igwidth}}%
    \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
    \global\renewcommand*{\LWR@igheightstyle}{height:\LWR@printlength{\LWR@igheight}}%
}{}%
```

Create the HTML reference with the graphicspath, filename, extension, alt tag, style, and class.

The `\LWR@origtilde` adds space between tags in case this is being done inside a `\savebox` where `\newline` has no effect.

```
\LWR@traceinfo{LWR@includegraphicsb: about to create href}%
\href{\LWR@parsedfilename}%
{% start of href
\LWR@traceinfo{LWR@includegraphicsb: about to LWR@htmltag}%
\LWR@htmltag{% start of image tags
img src="%
\begingroup\@sanitize\LWR@parsedfilename\endgroup%
" \LWR@orignewline%
```

Only include a style tag if a width, height, angle, or scale was given:

```
\ifthenelse{
    \NOT\equal{\LWR@igwidthstyle}{} \OR
    \NOT\equal{\LWR@igheightstyle}{} \OR
    \NOT\equal{\LWR@igorigin}{} \OR
    \NOT\equal{\LWR@igangle}{} \OR
    \NOT\equal{\LWR@igxscale}{1} \OR
    \NOT\equal{\LWR@igyscale}{1}
}%
{\LWR@origtilde{} style="%
\ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
{\LWR@igwidthstyle;}{}%
\ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
{\LWR@igheightstyle;}{}%
\ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
{\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}%
\ifthenelse{\NOT\equal{\LWR@igangle}{}}%
{%
\LWR@rotstyle{-ms-}{\LWR@igangle} % extra space
\LWR@rotstyle{-webkit-}{\LWR@igangle} % extra space
\LWR@rotstyle{}{\LWR@igangle %
}}{}%
\ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
\NOT\equal{\LWR@igyscale}{1}}%
{\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale} % extra space
\LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale} % extra space
\LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}}{} % extra space
%
\ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
{opacity:\LWR@opacity; }%
{}%
%
" \LWR@orignewline}{}%
```

Set the class:

```
\LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
}% end of image tags
}% end of href
```

Return to original page size and font size:

```
\endgroup
\LWR@traceinfo{LWR@includegraphicsb done}%
}
```

`\includegraphics` `[`⟨*key=val*⟩`]` `{`⟨*filename*⟩`}`

Handles width and height, converted to fixed width and heights.

The user should always use no file suffix in the document source.

```
\AtBeginDocument{

\LWR@traceinfo{Patching includegraphics.}

\LetLtxMacro\LWR@origincludegraphics\includegraphics

\renewcommand*{\includegraphics}
{%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
\LWR@traceinfo{includegraphics}%
\LWR@ensuredoingapar%
\LWR@includegraphicsb%
}% includegraphics
}% AtBeginDocument
```

§ 227.5 Boxes

`\LWR@rotboxorigin` Holds the origin key letters.

```
\newcommand*{\LWR@rotboxorigin}{}
```

`\LWR@originname` `{`⟨*letter*⟩`}`

Given one LaTeX origin key value, translate into an HTML origin word:

```
\newcommand*{\LWR@originname}[1]{%
\ifthenelse{\equal{#1}{t}}{top}{}%
\ifthenelse{\equal{#1}{b}}{bottom}{}%
\ifthenelse{\equal{#1}{c}}{center}{}%
\ifthenelse{\equal{#1}{l}}{left}{}%
\ifthenelse{\equal{#1}{r}}{right}{}%
}
```

`\LWR@originnames` `{`⟨*letters*⟩`}`

Given one- or two-letter LaTeX origin key values, translate into HTML origin words:

```
\newcommand*{\LWR@originnames}[1]{%
\StrChar{#1}{1}[\LWR@strresult]%
\LWR@originname{\LWR@strresult}
```

```
\StrChar{#1}{2}[\LWR@strresult]%
\LWR@originname{\LWR@strresult}
}
```

Handle the origin key for `\rotatebox`:

```
\define@key{krotbox}{origin}{%
\renewcommand*{\LWR@rotboxorigin}{#1}%
}
```

These keys are ignored:

```
\define@key{krotbox}{x}{}
\define@key{krotbox}{y}{}
\define@key{krotbox}{units}{}
```

`\rotatebox` `[`⟨*keyval list*⟩`]` `{`⟨*angle*⟩`}` `{`⟨*text*⟩`}`

```
\AtBeginDocument{
```

The HTML version:

```
\NewDocumentCommand{\LWR@HTML@rotatebox}{O{} m +m}{%
```

Reset the origin to “none-given”:

```
\renewcommand*{\LWR@rotboxorigin}{}
```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
\setkeys{krotbox}{#1}%
```

Select `inline-block` so that HTML will transform this span:

```
\LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
\ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}%
{transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%
```

Print the rotation information:

```
\LWR@rotstyle{-ms-}{#2} % extra space
\LWR@rotstyle{-webkit-}{#2} % extra space
\LWR@rotstyle{}{#2} % extra space
"}\LWR@orignewline%
```

Print the text to be rotated:

```
\begin{LWR@nestspan}%
#3%
```

Close the span:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
}
```

The high-level interface:

```
\LWR@formatted{rotatebox}

}% AtBeginDocument
```

\scalebox {⟨*h-scale*⟩} [⟨*v-scale*⟩] {⟨*text*⟩}

```
\AtBeginDocument{
```

The HTML version:

```
\NewDocumentCommand{\LWR@HTML@scalebox}{m o m}{%
```

Select `inline-block` so that HTML will transform this span:

```
\LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
\LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
\LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
\LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
"}%
```

Print the text to be scaled:

```
\begin{LWR@nestspan}%
#3%
```

Close the span:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
}
```

The high-level interface:

```
\LWR@formatted{scalebox}

}% AtBeginDocument
```

`\reflectbox` {⟨*text*⟩}

```
\AtBeginDocument{

\newcommand{\LWR@HTML@reflectbox}[1]{%
\scalebox{-1}[1]{#1}%
}% \reflectbox

\LWR@formatted{reflectbox}

}% AtBeginDocument
```

`\resizebox` {⟨*h-length*⟩} {⟨*v-length*⟩} {⟨*text*⟩}

Simply prints its text argument.

```
\AtBeginDocument{

\NewDocumentCommand{\LWR@HTML@resizebox}{s m m m}{%
#4%
}

\LWR@formatted{resizebox}

}% AtBeginDocument
```

File 136 `lwarp-graphicx.sty`

§ 228 Package graphicx

Pkg `graphicx` **graphicx** is emulated.

graphicx loads **graphics**, which also loads **lwarp-graphics**, which remembers the original graphics definitions for use inside a `lateximage`, and then patches them `\AtBeginDocument` for HTML output.

lwarp-graphics handles the syntax of either **graphics** or **graphicx**.

for HTML output:

```
\LWR@ProvidesPackagePass{graphicx}[2017/06/01]
```

File 137 `lwarp-grffile.sty`

§ 229 Package grffile

Pkg grffile

⚠ matching PDF and SVG

grffile is supported as-is. File types known to the browser are displayed, and unknown file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

lwarp-grffile now exists as a placeholder since **grffile** used to be emulated by **lwarp**, and thus older versions of **lwarp-grffile** may exist and should be overwritten by this newer version.

for HTML output:

```
\LWR@ProvidesPackagePass{grffile}
```

File 138 `lwarp-grid.sty`

§ 230 Package grid

Pkg grid

grid is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{grid}
\newenvironment*{gridenv}{}{}
```

File 139 `lwarp-grid-system.sty`

§ 231 Package grid-system

(*Emulates or patches code by* Marcus Bitzl.)

Pkg grid-system

grid-system is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{grid-system}[2014/02/16]
```

(`\ifdef` is in case the older syntax is removed.)

```
\AtBeginEnvironment{Row}{\setlength{\linewidth}{6in}}

```

```
\ifdef{\endrow}{
    \AtBeginEnvironment{row}{\setlength{\linewidth}{6in}}
}{}

\renewcommand{\gridsystem@finishcell}{\hspace{\gridsystem@cellsep}}
```

File 140 `lwarp-gridset.sty`

§ 232 Package gridset

Pkg `gridset` **gridset** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{gridset}

\newcommand*{\gridbase}{}
\newcommand*{\gridinterval}{}
\newcommand*{\savepos}[1]{}
\newcounter{gridcnt}
\newcommand*{\vskipnextgrid}{}
\newcommand*{\thegridinfo}[1]{}
\newcommand*{\theposinfo}[1]{}
\newcommand*{\theypos}[1]{}
```

File 141 `lwarp-hang.sty`

§ 233 Package hang

(*Emulates or patches code by* Andreas Nolda.)

Pkg `hang` **hang** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{hang}

\newlength{\hangingindent}
\setlength{\hangingindent}{1em}
\newlength{\hangingleftmargin}
\setlength{\hangingleftmargin}{0em}

\newcommand*{\LWR@findhangingleftmargin}{%
\setlength{\LWR@templengthone}{\hangingleftmargin}%
\addtolength{\LWR@templengthone}{\hangingindent}%
}

```

```
\newenvironment{hangingpar}
{
    \LWR@findhangingleftmargin%
    \BlockClass[%
        \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
        \LWR@print@mbox{text-indent:-\LWR@printlength{\hangingindent}}%
    ]%
    {hangingpar}%
}
{\endBlockClass}

\newenvironment{hanginglist}
{%
    \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}%
    \renewcommand*{\LWR@printopenlist}{%
        \LWR@findhangingleftmargin%
        ul style="%
            \LWR@print@mbox{list-style-type:none;} % extra space
            \LWR@print@mbox{%
                margin-left:\LWR@printlength{\LWR@templengthone}%
            } ; % extra space
            \LWR@print@mbox{%
                text-indent:-\LWR@printlength{\hangingindent}%
            }%
        "%
    }%
    \let\item\LWR@itemizeitem%
    \list{}{}%
}
{\endlist}

\newenvironment{compacthang}
{\hanginglist}
{\endhanginglist}

\newlength{\labeledleftmargin}
\setlength{\labeledleftmargin}{0em}

\newenvironment{labeledpar}[2]
{%
    \BlockClass[%
        \LWR@findhangingleftmargin%
        \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
        \LWR@print@mbox{text-indent:-\LWR@printlength{\hangingindent}}%
    ]{labeledpar}#2%
}
{\endBlockClass}

\newenvironment{labeledlist}[1]
{\hanginglist}
```

```
{\endhanginglist}

\newenvironment{compactlabel}[1]
{\hanginglist}
{\endhanginglist}
```

File 142 `lwarp-hanging.sty`

§ 234 Package hanging

Pkg `hanging` **hanging** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{hanging}

\@ifclassloaded{memoir}{
\let\hangpara\relax
\let\hangparas\relax
\let\endhangparas\relax
\let\hangpunct\relax
\let\endhangpunct\relax
}{}
```

`\hangpara` {⟨*indent*⟩} {⟨*afternum*⟩}

Use `hangparas` instead.

```
\newcommand*{\hangpara}[2]{}
```

Env `hangparas` {⟨*indent*⟩} {⟨*afternum*⟩}

```
\newenvironment*{hangparas}[2]
{%
    \BlockClass[%
        \LWR@print@mbox{margin-left:\LWR@printlength{#1}} ; %
        \LWR@print@mbox{text-indent:-\LWR@printlength{#1}}%
    ]%
    {hangingpar}%
}
{\endBlockClass}
```

Env `hangpunct`

```
\newenvironment*{hangpunct}
{\BlockClass{hangpunct}}
{\endBlockClass}
```

```
\newcommand{\nhpt}{.}
\newcommand{\nhlq}{‘}
\newcommand{\nhrq}{’}
```

File 143 `lwarp-hypcap.sty`

§ 235 Package hypcap

Pkg hypcap **hypcap** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{hypcap}

\newcommand*{\capstart}{}
\newcommand*{\hypcapspace}{}
\newcommand*{\hypcapredef}[1]{}
\newcommand*{\capstartfalse}{}
\newcommand*{\capstarttrue}{}
```

File 144 `lwarp-hypdestopt.sty`

§ 236 Package hypdestopt

Pkg hypdestopt **hypdestopt** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{hypdestopt}
```

File 145 `lwarp-hypernat.sty`

§ 237 Package hypernat

Pkg hypernat **hypernat** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{hypernat}
```

File 146 `lwarp-hyperref.sty`

§ 238 Package hyperref

(*Emulates or patches code by* Sebastian Rahtz, Heiko Oberdiek.)

Pkg hyperref **hyperref** is emulated.

for HTML output:

```
% \LWR@ProvidesPackageDrop{hyperref}
\typeout{Using the lwarp html version of package ‘hyperref’, discarding options.}
\typeout{    Are not using ProvidesPackage, so that other packages}
\typeout{    do not attempt to patch lwarp’s version of ‘hyperref’.}
% \ProvidesPackage{lwarp-#1-#2}
\DeclareOption*{}
\ProcessOptions\relax

\newcommand*{\hypersetup}[1]{}
\newcommand*{\hyperbaseurl}[1]{}
```

\hyperimage {⟨*URL*⟩} {⟨*alt text*⟩}

Insert an image with alt text:

```
\NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
\LWR@ensuredoingapar%
\def\LWR@templink{#1}%
\@onelevel@sanitize\LWR@templink%
\LWR@htmltag{img src="\LWR@templink" alt="#2" class="hyperimage"}%
\LWR@ensuredoingapar%
\endgroup%
}

\newrobustcmd*{\hyperimage}{%
\begingroup%
\catcode‘\#=12%
\catcode‘\%=12%
\catcode‘\&=12%
\catcode‘\~=12%
\catcode‘\_=12%
\LWR@hyperimageb%
}

```

\hyperdef {⟨*1: category*⟩} {⟨*2: name*⟩} {⟨*3: text*⟩}

Creates an HTML anchor to `category.name` with the given text.

```
\NewDocumentCommand{\LWR@hyperdefb}{m m +m}{%
\LWR@ensuredoingapar%
\LWR@sublabel{#1.#2}%
#3%
\endgroup%
}

\newcommand*{\hyperdef}{%
\begingroup%
```

```
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@hyperdefb%
}

```

`\LWR@hyperrefb` {⟨*1: URL*⟩} {⟨*2: category*⟩} {⟨*3: name*⟩} {⟨*4: text*⟩}

Creates an HTML link to `URL#category.name` with the given text.

```
\newcommand{\LWR@hyperreffinish}[1]{%
\begingroup%
\RenewDocumentCommand{\ref}{}{\LWR@ref@ignorestar}%
#1%
\endgroup%
\LWR@htmltag{/a}%
}

\newcommand*{\LWR@hyperrefbb}[3]{%
\LWR@htmltag{%
    a href="%
        \detokenize\expandafter{#1}\LWR@hashmark%
        \detokenize\expandafter{#2}.\detokenize\expandafter{#3}%
    "%
}%
\endgroup%
\LWR@hyperreffinish%
}

\newrobustcmd*{\LWR@hyperrefb}{%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@hyperrefbb%
}
```

`\LWR@hyperrefc` [⟨*label*⟩] {⟨*text*⟩}

Creates text as an HTML link to the LaTeX label.

```

\NewDocumentCommand{\LWR@hyperrefcb}{O{label}}{%
\LWR@startref{#1}%
\endgroup%
```

```
\LWR@hyperreffinish%
}

\newcommand*{\LWR@hyperrefc}{%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@hyperrefcb%
}
```

`\hyperref` {⟨*1: URL*⟩} {⟨*2: category*⟩} {⟨*3: name*⟩} {⟨*4: text*⟩} —or— [⟨*1: label*⟩] {⟨*2: text*⟩}

```
\DeclareRobustCommand*{\hyperref}{%
\LWR@ensuredoingapar%
\@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
}
```

`\hypertarget` {⟨*name*⟩} {⟨*text*⟩}

Creates an anchor to `name` with the given text.

```
\NewDocumentCommand{\LWR@hypertargetb}{m +m}{%
\label{LWR-ht-#1}%
#2%
\endgroup%
}

\newcommand*{\hypertarget}{%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@hypertargetb%
}
```

`\hyperlink` {⟨*name*⟩} {⟨*text*⟩}

Creates a link to the anchor created by hypertarget, with the given link text.

Declared because also defined by **memoir**.

```
\DeclareDocumentCommand{\LWR@hyperlinkb}{m}{%
\LWR@hyperrefcb[LWR-ht-#1]%
}
```

```
\DeclareDocumentCommand{\hyperlink}{}{%
\LWR@ensuredoingapar%
\begingroup%
\catcode`\#=12%
\catcode`\%=12%
\catcode`\&=12%
\catcode`\~=12%
\catcode`\_=12%
\LWR@hyperlinkb%
}
```

`\autoref` * {⟨*label*⟩}

For HTML, `\cleveref` is used instead.

```
\NewDocumentCommand{\autoref}{s m}{%
\IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
}
```

`\autopageref` {⟨*label*⟩}

For HTML, `\cleveref` is used instead.

```
\NewDocumentCommand{\autopageref}{s m}{%
\IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
}
```

`\pdfstringdef` {⟨*macroname*⟩} {⟨*TEXstring*⟩}

```
\newcommand{\pdfstringdef}[2]{}
```

`\pdfbookmark` [⟨*level*⟩] {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\pdfbookmark}[3][]{}
```

`\currentpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\currentpdfbookmark}[2]{}
```

`\subpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\subpdfbookmark}[2]{}
```

`\belowpdfbookmark` {⟨*text*⟩} {⟨*name*⟩}

```
\newcommand{\belowpdfbookmark}[2]{}
```

\texorpdfstring {⟨*TEXstring*⟩} {⟨*PDFstring*⟩}

```
\newcommand{\texorpdfstring}[2]{#1}
```

\hypercalcbp {⟨*dimen*⟩} From **hyperref**.

```
\def\hypercalcbp#1{%
\strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
}%
```

\Acrobatmenu {⟨*menuoption*⟩} {⟨*text*⟩}

```
\newcommand{\Acrobatmenu}[2]{}
```

\TextField [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\TextField}[2][]{}
```

\CheckBox [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\CheckBox}[2][]{}
```

\ChoiceMenu [⟨*parameters*⟩] {⟨*label*⟩} {⟨*choices*⟩}

```
\DeclareRobustCommand{\ChoiceMenu}[3][]{}
```

\PushButton [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\PushButton}[2][]{}
```

\Submit [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Submit}[2][]{}
```

\Reset [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Reset}[2][]{}
```

\Gauge [⟨*parameters*⟩] {⟨*label*⟩}

```
\DeclareRobustCommand{\Gauge}[2][]{}
```

\LayoutTextField {⟨*label*⟩} {⟨*field*⟩}

```
\newcommand*{\LayoutTextField}[2]{}
```

\LayoutChoiceField {⟨*label*⟩} {⟨*field*⟩}

```
147 \newcommand*{\LayoutChoiceField}[2]{}
```

\LayoutCheckField {⟨*label*⟩} {⟨*field*⟩}

```
148 \newcommand*{\LayoutCheckField}[2]{}
```

\MakeRadioField {⟨*width*⟩} {⟨*height*⟩}

```
149 \newcommand*{\MakeRadioField}[2]{}
```

\MakeCheckField {⟨*width*⟩} {⟨*height*⟩}

```
150 \newcommand*{\MakeCheckField}[2]{}
```

\MakeTextField {⟨*width*⟩} {⟨*height*⟩}

```
151 \newcommand*{\MakeTextField}[2]{}
```

\MakeChoiceField {⟨*width*⟩} {⟨*height*⟩}

```
152 \newcommand*{\MakeChoiceField}[2]{}
```

\MakeFieldButton {⟨*text*⟩}

```
153 \newcommand{\MakeFieldButton}[1]{}
```

File 147 `lwarp-hyperxmp.sty`

§ 239 Package

hyperxmp

Pkg `hyperxmp` Emulated.

for HTML output: Discard all options for **lwarp-hyperxmp**:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 148 `lwarp-hyphenat.sty`

§ 240 Package

hyphenat

Pkg hyphenat

hyphenat is emulated during HTML output, while the print-mode version is used inside a `lateximage`.

for HTML output:

```
\LWR@ProvidesPackagePass{hyphenat}

\LetLtxMacro\LWRHYNAT@origtextnhtt\textnhtt
\LetLtxMacro\LWRHYNAT@orignhttfamily\nhttfamily
\LetLtxMacro\LWRHYNAT@orignohyphens\nohyphens
\LetLtxMacro\LWRHYNAT@origbshyp\bshyp
\LetLtxMacro\LWRHYNAT@origfshyp\fshyp
\LetLtxMacro\LWRHYNAT@origdothyp\dothyp
\LetLtxMacro\LWRHYNAT@origcolonhyp\colonhyp
\LetLtxMacro\LWRHYNAT@orighyp\hyp

\LetLtxMacro\textnhtt\texttt
\LetLtxMacro\nhttfamily\ttfamily

\renewcommand{\nohyphens}[1]{#1}
\renewrobustcmd{\bshyp}{%
    \ifmmode\backslash\else\textbackslash\fi%
}
\renewrobustcmd{\fshyp}{/}
\renewrobustcmd{\dothyp}{.}
\renewrobustcmd{\colonhyp}{:}
\renewrobustcmd{\hyp}{-}

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\textnhtt\LWRHYNAT@origtextnhtt%
\LetLtxMacro\nhttfamily\LWRHYNAT@orignhttfamily%
\LetLtxMacro\nohyphens\LWRHYNAT@orignohyphens%
\LetLtxMacro\bshyp\LWRHYNAT@origbshyp%
\LetLtxMacro\fshyp\LWRHYNAT@origfshyp%
\LetLtxMacro\dothyp\LWRHYNAT@origdothyp%
\LetLtxMacro\colonhyp\LWRHYNAT@origcolonhyp%
\LetLtxMacro\hyp\LWRHYNAT@orighyp%
}
```

File 149 `lwarp-idxlayout.sty`

§ 241 Package idxlayout

(*Emulates or patches code by* Thomas Titz.)

Pkg `idxlayout` Emulated.

for HTML output: Discard all options for **lwarp-idxlayout**:

```
\LWR@ProvidesPackageDrop{idxlayout}

\newcommand{\LWR@indexprenote}{}

\preto\printindex{

\LWR@orignewpage
\LWR@startpars

\LWR@indexprenote

}

\newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
\newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{}}

\newcommand{\idxlayout}[1]{}
\newcommand*{\indexfont}{}
\newcommand*{\indexjustific}{}
\newcommand*{\indexsubsdelim}{}
\newcommand*{\indexstheadcase}{}
```

File 150 `lwarp-ifoddpage.sty`

§ 242 Package ifoddpage

(*Emulates or patches code by* Martin Scharrer.)

Pkg `ifoddpage` **ifoddpage** is emulated.

for HTML output: Discard all options for **lwarp-ifoddpage**:

```
\LWR@ProvidesPackageDrop{ifoddpage}
```

```
\newif\ifoddpage

\newif\ifoddpageoroneside

\DeclareRobustCommand{\checkoddpage}{\oddpagetrue\oddpageoronesidetrue}

\def\oddpage@page{1}

\def\@ifoddpage{%
        \expandafter\@firstoftwo
}

\def\@ifoddpageoroneside{%
        \expandafter\@firstoftwo
}
```

File 151 `lwarp-imakeidx.sty`

§ 243 Package imakeidx

(*Emulates or patches code by* Enrico Gregorio.)

Pkg imakeidx **imakeidx** is patched for use by **lwarp**.

letter headings When using **makeindex**, to match the print and HTML output's display of index letter headings, specify the `lwarp.ist` style:

```
\makeindex[options={-s lwarp.ist}]
```

(For HTML the `lwarp.ist` style is used automatically, which displays letter headings. When using **xindy** the default style also displays letter headings.)

index setup See section 9.5.15 for how to setup **lwarpmk** to process the indexes with **imakeidx**, both with and without shell escape.

for HTML output:

```
\LWR@ProvidesPackagePass{imakeidx}
```

Use the new HTML suffix:

```
\catcode`\_=12%
\define@key{imki}{name}{\def\imki@name{#1_html}}
\catcode`\_=8%
```

\printindex The HTML version of \printindex:

```
\catcode`\_=12%
```

```
\renewcommand*{\printindex}[1][\imki@jobname]{%
\LWR@orignewpage%
\LWR@startpars%
\ifstrequal{#1}{\imki@jobname}{%
  \@ifundefined{#1@idxfile}{%
        \imki@error{#1}%
    }{%
        \imki@putindex{#1}%
    }%
}{%
  \@ifundefined{#1_html@idxfile}{\imki@error{#1_html}}{\imki@putindex{#1_html}}%
}%
}

\catcode`\_=8%
```

\@index The HTML version of \index:

```
\catcode`\_=12%

\def\@index[#1]{%
    \ifstrequal{#1}{\imki@jobname}%
    {%
        \@ifundefined{#1@idxfile}%
        {%
            \PackageWarning{imakeidx}{Undefined index file `#1'}%
            \begingroup
            \@sanitize
            \imki@nowrindex%
        }%
        {%
            \edef\@idxfile{#1}%
            \begingroup
            \@sanitize
            \@wrindex\@idxfile%
        }%
    }%
    {%
        \@ifundefined{#1_html@idxfile}%
        {%
            \PackageWarning{imakeidx}{Undefined index file `#1_html'}%
            \begingroup
            \@sanitize
            \imki@nowrindex%
        }%
        {%
            \edef\@idxfile{#1_html}%
            \begingroup
```

```
            \@sanitize
            \@wrindex\@idxfile%
        }%
    }%
}

\catcode`\_=8%
```

\item
\subitem
\subsubitem

HTML versions of \item, etc.:

```
\appto\theindex{%
    \let\item\LWR@indexitem%
    \let\subitem\LWR@indexsubitem%
    \let\subsubitem\LWR@indexsubsubitem%
}
```

\imki@wrindexentrysplit {⟨*file*⟩} {⟨*entry*⟩} {⟨*page*⟩}
\imki@wrindexentryunique {⟨*file*⟩} {⟨*entry*⟩} {⟨*page*⟩}

While writing index entries, adds an HTML label, and writes the label's index instead of the page number:

```
\renewcommand\imki@wrindexentrysplit[3]{%
\addtocounter{LWR@autoindex}{1}%
\LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
  \expandafter\protected@write\csname#1@idxfile\endcsname{}%
    {\string\indexentry{#2}{\arabic{LWR@autoindex}}}%
}

\renewcommand\imki@wrindexentryunique[3]{%
\addtocounter{LWR@autoindex}{1}%
\LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
  \protected@write\@indexfile{}%
    {\string\indexentry[#1]{#2}{\arabic{LWR@autoindex}}}%
}

\def\imki@wrindexsplit#1#2{%
\imki@wrindexentrysplit{#1}{#2}{\thepage}%
\endgroup\imki@showidxentry{#1}{#2}%
  \@esphack%
}

\def\imki@wrindexunique#1#2{%
\imki@wrindexentryunique{#1}{#2}{\thepage}%
\endgroup\imki@showidxentry{#1}{#2}%
  \@esphack%
  }
```

```
89
```

`\LWR@imki@setxdydefopts`

Sets the **xindy** HTML options, ignoring the user's settings.

```
90 \newcommand*{\LWR@imki@setxdydefopts}{%
91     \edef\imki@options{ \space %
92         -M \space \LWR@xindyStyle\space %
93         -L \space \LWR@xindyLanguage\space %
94         -C \space \LWR@xindyCodepage\space %
95     }%
96 }
```

`\LWR@imki@setdefopts` `{`⟨*user options*⟩`}`

Sets the HTML options, added to the user's settings, depending on whether **makeindex** or **xindy** are used.

For **makeindex**, the user's choice is ignored, and only the **lwarp** version is used. (Only one style at a time is possible.)

For **xindy**, multiple modules may be specified, and the **lwarp** version is appended.

```
97 \newcommand*{\LWR@imki@setdefopts}[1]{%
98 \ifblank{#1}{%
99     \edef\imki@options{\space -s \space \LWR@makeindexStyle \space}%
100    \ifdefstring{\imki@progdefault}{xindy}{\LWR@imki@setxdydefopts}{}%
101    \ifdefstring{\imki@progdefault}{texindy}{\LWR@imki@setxdydefopts}{}%
102    \ifdefstring{\imki@progdefault}{truexindy}{\LWR@imki@setxdydefopts}{}%
103 }{%
104    \edef\imki@options{\space #1 \space}%
105 }%
106 }
```

`\imki@makeindex` Use the new HTML options:

```
107 \xpatchcmd{\imki@makeindex}
108     {\let\imki@options\space}
109     {\LWR@imki@setdefopts{}}%
110     {}
111     {\LWR@patcherror{imakeidx}{makeindex}}
```

Use the new HTML options.

```
112 \define@key{imki}{options}{\LWR@imki@setdefopts{#1}}
```

`\imki@resetdefaults` Use the new HTML options:

```
113 \xpatchcmd{\imki@resetdefaults}
114     {\def\imki@options{ }}
```

```
    {\LWR@imki@setdefopts{}}
    {}
    {\LWR@patcherror{imakeidx}{resetdefaults}}
```

`theindex` was already defined `\AtBeginDocument` by the **lwarp** core, so it must be redefined here similarly, but patched for **imakeidx**:

Env `theindex`

```
\AtBeginDocument{
\renewenvironment*{theindex}{%
    \imki@maybeaddtotoc
     \imki@indexlevel{\indexname}
    \let\item\LWR@indexitem%
    \let\subitem\LWR@indexsubitem%
    \let\subsubitem\LWR@indexsubsubitem%
}{}
}% AtBeginDocument
```

Update to the new defaults:

```
\imki@resetdefaults
```

Update to the new patches:

`\AtBeginDocument` is because `\@wrindex` is previously defined as `\AtBeginDocument` in the **lwarp** core.

```
\ifimki@splitindex
  \let\imki@startidx\imki@startidxunique
  \AtBeginDocument{\let\@wrindex\imki@wrindexunique}
  \let\imki@putindex\imki@putindexunique
  \let\imki@wrindexentry\imki@wrindexentryunique
  \let\imki@startidxsplit\@undefined
  \let\imki@wrindexsplit\@undefined
  \let\imki@putindexsplit\@undefined
\else
  \let\imki@startidx\imki@startidxsplit
  \AtBeginDocument{\let\@wrindex\imki@wrindexsplit}
  \let\imki@putindex\imki@putindexsplit
  \let\imki@wrindexentry\imki@wrindexentrysplit
  \let\imki@startidxunique\@undefined
  \let\imki@wrindexunique\@undefined
  \let\imki@putindexunique\@undefined
\fi
```

File 152 `lwarp-indentfirst.sty`

§ 244 Package indentfirst

Pkg `indentfirst` **indentfirst** is ignored.

Discard all options for **lwarp-indentfirst**:

for HTML output:
```
\LWR@ProvidesPackageDrop{indentfirst}
```

File 153 `lwarp-index.sty`

§ 245 Package index

(*Emulates or patches code by* David M. Jones.)

Pkg `index` **index** is patched for use by **lwarp**.

for HTML output:
```
\LWR@ProvidesPackagePass{index}
```

Use `\theLWR@autoindex` instead of `\thepage`. `\@tempswatrue` is used to force an immediate write to the index file instead of waiting until the end of the page.

```
\xpatchcmd{\newindex}
    {\x@newindex[thepage]}
    {%
        \@tempswatrue%
        \x@newindex[theLWR@autoindex]%
    }
    {}
    {\LWR@patcherror{index}{newindex}}

\xpatchcmd{\renewindex}
    {\x@renewindex[thepage]}
    {%
        \@tempswatrue%
        \x@renewindex[theLWR@autoindex]%
    }
    {}
    {\LWR@patcherror{index}{renewindex}}
```

Patched to set a new autoindex:

```
\xpatchcmd{\@wrindex}
    {\begingroup}
    {%
        \addtocounter{LWR@autoindex}{1}%                    lwarp
        \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%    lwarp
        \begingroup%
    }
    {}
    {\LWR@patcherror{index}{@wrindex}}
```

`\AtBeginDocument` **lwarp** core `\lets` `\@wrindex` to `\LWR@wrindex`. Since the **index** package has been loaded, `\let` to its version instead:

```
\let\LWR@index@wrindex\@wrindex

\AtBeginDocument{
\let\@wrindex\LWR@index@wrindex
}
```

Modified to add `\index@prologue`:

```
\AtBeginDocument{
\renewenvironment*{theindex}{%
    \LWR@indexsection{\indexname}%
    \ifx\index@prologue\@empty\else
        \index@prologue
        \bigskip
    \fi
    \let\item\LWR@indexitem%
    \let\subitem\LWR@indexsubitem%
    \let\subsubitem\LWR@indexsubsubitem%
}{}
}% AtBeginDocument
```

Disabled:

```
\def\@showidx#1{}
\let\@texttop\relax
\renewcommand*{\raggedbottom}{}
\renewcommand*{\flushbottom}{}
\renewcommand*{\markboth}[2]{}
\renewcommand*{\markright}[1]{}
```

File 154 `lwarp-inputenc.sty`

§ 246 Package inputenc

Pkg `inputenc` Error if **inputenc** is loaded after **lwarp**.

for HTML output:

```
\LWR@ProvidesPackageDrop{inputenc}
\LWR@loadbefore{inputenc}
```

File 155 `lwarp-inputenx.sty`

§ 247 Package inputenx

Pkg `inputenx` Error if **inputenx** is loaded after **lwarp**.

for HTML output:

```
\LWR@ProvidesPackageDrop{inputenx}
\LWR@loadbefore{inputenx}
```

File 156 `lwarp-intopdf.sty`

§ 248 Package intopdf

Pkg `intopdf` **intopdf** is emulated.

The MIME type and description are ignored for now.

for HTML output:

```
\LWR@ProvidesPackageDrop{intopdf}

\NewDocumentCommand{\attachandlink}{m o m m}{%
    \href{#1}{#4}%
}
```

File 157 `lwarp-keyfloat.sty`

§ 249 Package keyfloat

(*Emulates or patches code by* Brian Dunn.)

Pkg `keyfloat` **keyfloat** is supported with minor adjustments.

⚠ `keywrap` If placing a `\keyfig[H]` inside a `keywrap`, use an absolute width for `\keyfig`, instead of `lw`-proportional widths. (The `[H]` option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the `lw` option.)

for HTML output:

```
\LWR@ProvidesPackagePass{keyfloat}
```

After **keyfloat** has loaded:

```
\AtBeginDocument{

\RenewDocumentCommand{\KFLT@onefigureimage}{}
{%
\LWR@traceinfo{KFLT@onefigureimage}%
% \begin{lrbox}{\KFLT@envbox}%
\ifthenelse{\NOT\equal{\KFLT@lw}{}}%
{\includegraphics%
[scale=\KFLT@s,width=\KFLT@imagewidth]{\KFLT@i}}%
{% not linewidth
\ifthenelse{\dimtest{\KFLT@w}{>}{0pt}}%
{% width is given
\ifthenelse{\dimtest{\KFLT@h}{>}{0pt}}%
{% w and h
\includegraphics%
[scale=\KFLT@s,%
width=\KFLT@imagewidth,height=\KFLT@h]{\KFLT@i}%
}% w and h
{% only w
\includegraphics%
[scale=\KFLT@s,width=\KFLT@imagewidth]{\KFLT@i}%
}% only w
}% width is given
{% width is not given
\ifthenelse{\dimtest{\KFLT@h}{>}{0pt}}%
{\includegraphics%
[scale=\KFLT@s,height=\KFLT@h]{\KFLT@i}}%
{\includegraphics%
```

```
[scale=\KFLT@s]{\KFLT@i}}%
}% width is not given
}% not linewidth
% \end{lrbox}%
% \unskip%
% \KFLT@findenvboxwidth%
% \begin{turn}{\KFLT@r}%
% \KFLT@frame{\usebox{\KFLT@envbox}}%
% \unskip%
% \end{turn}%
\LWR@traceinfo{KFLT@onefigureimage: done}%
}

\RenewDocumentEnvironment{KFLT@boxinner}{}
{%
\LWR@traceinfo{KFLT@boxinner}%
\LWR@stoppars%
}
{
\LWR@startpars%
\LWR@traceinfo{KFLT@boxinner: done}%
}


\DeclareDocumentEnvironment{KFLT@marginfloat}{O{-1.2ex} m}
{%
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}

\DeclareDocumentEnvironment{marginfigure}{o}
  {\begin{KFLT@marginfloat}{figure}}
  {\end{KFLT@marginfloat}}

\DeclareDocumentEnvironment{margintable}{o}
  {\begin{KFLT@marginfloat}{table}}
  {\end{KFLT@marginfloat}}

\DeclareDocumentEnvironment{keywrap}{m +m}
{%
\LWR@ensuredoingapar%
\setlength{\LWR@templengthone}{#1}%
\begin{LWR@BlockClassWP}{%
    float:right; width:\LWR@printlength{\LWR@templengthone};  % extra space
    margin:10pt%
}%
```

```
{%
    width:\LWR@printlength{\LWR@templengthone}%
}%
{marginblock}%
\setlength{\linewidth}{.95\LWR@templengthone}%
#2%
\end{LWR@BlockClassWP}%
}
{%
}

}% AtBeginDocument
```

File 158 `lwarp-layaureo.sty`

§ 250 Package layaureo

Pkg `layaureo` **layaureo** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{layaureo}[2004/09/16]
```

File 159 `lwarp-layout.sty`

§ 251 Package layout

(*Emulates or patches code by* Kent McPherson, Johannes Braams, Hideo Umeki.)

Pkg `layout` **layout** is emulated.

for HTML output: Discard all options for **lwarp-layout**:

```
\LWR@ProvidesPackageDrop{layout}

\NewDocumentCommand{\layout}{s}{}
```

File 160 `lwarp-leading.sty`

§ 252 Package leading

Pkg `leading` **leading** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{leading}[2008/12/11]
```

```
\newcommand\leading[1]{}
```

File 161 `lwarp-letterspace.sty`

§ 253 Package letterspace

(*Emulates or patches code by* R Schlicht.)

Pkg `letterspace` **letterspace** is a subset of microtype, which is pre-loaded by **lwarp**. All user options and macros are ignored and disabled.

for HTML output: Discard all options for **lwarp-letterspace**:

```
\LWR@ProvidesPackageDrop{letterspace}

\newcommand*\lsstyle{}
\newcommand\textls[2][]{}
\def\textls#1#{}
\newcommand*\lslig[1]{#1}
```

File 162 `lwarp-lettrine.sty`

§ 254 Package lettrine

(*Emulates or patches code by* Daniel Flipo.)

Pkg `lettrine` Emulated.

for HTML output: Discard all options for **lwarp-lettrine**:

```
\LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a `<span>` of class `lettrine`, and the following text is in a `<span>` of class `lettrinetext`. `\lettrine [`⟨*keys*⟩`] {`⟨*letter*⟩`} {`⟨*additional text*⟩`}`

```
\DeclareDocumentCommand{\lettrine}{o m m}{%
\InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} % extra space
}

\newcounter{DefaultLines}
\setcounter{DefaultLines}{2}
\newcounter{DefaultDepth}
\newcommand*{\DefaultOptionsFile}{\relax}
\newcommand*{\DefaultLoversize}{0}
```

```
\newcommand*{\DefaultLraise}{0}
\newcommand*{\DefaultLhang}{0}
\newdimen\DefaultFindent
\setlength{\DefaultFindent}{\z@}
\newdimen\DefaultNindent
\setlength{\DefaultNindent}{0.5em}
\newdimen\DefaultSlope
\setlength{\DefaultSlope}{\z@}
\newdimen\DiscardVskip
\setlength{\DiscardVskip}{0.2\p@}
\newif\ifLettrineImage
\newif\ifLettrineOnGrid
\newif\ifLettrineRealHeight

\newcommand*{\LettrineTextFont}{\scshape}
\newcommand*{\LettrineFontHook}{}
\newcommand*{\LettrineFont}[1]{\InlineClass{lettrine}{#1}}
\newcommand*{\LettrineFontEPS}[1]{\includegraphics[height=1.5ex]{#1}}
```

File 163 `lwarp-lineno.sty`

§ 255 Package lineno

(*Emulates or patches code by* Stephan I. Böttcher.)

Pkg `lineno` **lineno** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{lineno}

\newcommand*\resetlinenumber[1][\@ne]{}

\def\linenumbers{%
    \@ifnextchar[{\resetlinenumber}%]
                 {\@ifstar{\resetlinenumber}{}}%
    }

\newcommand*{\nolinenumbers}{}

\@namedef{linenumbers*}{\par\linenumbers*}
\@namedef{runninglinenumbers*}{\par\runninglinenumbers*}

\def\endlinenumbers{\par}
\let\endrunninglinenumbers\endlinenumbers
\let\endpagewiselinenumbers\endlinenumbers
\expandafter\let\csname endlinenumbers*\endcsname\endlinenumbers
\expandafter\let\csname endrunninglinenumbers*\endcsname\endlinenumbers
\let\endnolinenumbers\endlinenumbers
```

```
20
21 \def\pagewiselinenumbers{\linenumbers\setpagewiselinenumbers}
22
23 \def\runninglinenumbers{\setrunninglinenumbers\linenumbers}
24
25 \def\setpagewiselinenumbers{}
26
27 \def\setrunninglinenumbers{}
28
29 \def\linenomath{}%
30 \@namedef{linenomath*}{}%
31 \def\endlinenomath{}
32 \expandafter\let\csname endlinenomath*\endcsname\endlinenomath
33
34 \let\linelabel\label
35
36 \def\switchlinenumbers{\@ifstar{}{}}
37 \def\setmakelinenumbers#1{\@ifstar{}{}}
38
39 \def\leftlinenumbers{\@ifstar{}{}}
40 \def\rightlinenumbers{\@ifstar{}{}}
41
42 \newcounter{linenumber}
43 \newcount\c@pagewiselinenumber
44 \let\c@runninglinenumber\c@linenumber
45
46 \def\runningpagewiselinenumbers{}
47 \def\realpagewiselinenumbers{}
48
49
50 \NewDocumentCommand\modulolinenumbers{s o}{}
51
52 \chardef\c@linenumbermodulo=5
53 \modulolinenumbers[1]
54
55 \newcommand*\firstlinenumber[1]{}
56
57 \newcommand\internallinenumbers{}
58 \let\endinternallinenumbers\endlinenumbers
59 \@namedef{internallinenumbers*}{\internallinenumbers*}
60 \expandafter\let\csname endinternallinenumbers*\endcsname\endlinenumbers
61
62 \newcommand*{\linenoplaceholder}[1]{% redefine per language
63     (line number reference for \detokenize\expandafter{#1})
64 }
65
66 \newcommand*{\lineref}[2][]{\linenoplaceholder{#2}}
67 \newcommand*{\linerefp}[2][]{\linenoplaceholder{#2}}
68 \newcommand*{\linerefr}[2][]{\linenoplaceholder{#2}}
69
```

```
\newcommand\quotelinenumbers
   {\@ifstar\linenumbers{\@ifnextchar[\linenumbers{\linenumbers*}}}

\newdimen\linenumbersep
\newdimen\linenumberwidth
\newdimen\quotelinenumbersep

\quotelinenumbersep=\linenumbersep
\let\quotelinenumberfont\linenumberfont

\def\linenumberfont{\normalfont\tiny\sffamily}


\linenumberwidth=10pt
\linenumbersep=10pt

\def\thelinenumber{}

\def\LineNumber{}
\def\makeLineNumber{}
\def\makeLineNumberLeft{}
\def\makeLineNumberRight{}
\def\makeLineNumberOdd{}
\def\makeLineNumberEven{}
\def\makeLineNumberRunning{}


\newenvironment{numquote}     {\quote}{\endquote}
\newenvironment{numquotation} {\quotation}{\endquotation}
\newenvironment{numquote*}    {\quote}{\endquote}
\newenvironment{numquotation*}{\quotation}{\endquotation}

\newdimen\bframerule
\bframerule=\fboxrule

\newdimen\bframesep
\bframesep=\fboxsep

\newenvironment{bframe}
{%
    \LWR@forceminwidth{\bframerule}%
    \BlockClass[
        border:\LWR@printlength{\LWR@atleastonept} solid black ; %
        padding:\LWR@printlength{\bframesep}%
    ]{bframe}
}
{\endBlockClass}
```

File 164 `lwarp-lips.sty`

§ 256 Package lips

(*Emulates or patches code by* Matt Swift.)

Pkg `lips` **lips** is emulated.

```
% \LWR@ProvidesPackageDrop{lips}
\PackageInfo{lwarp}{Using the lwarp version of package ‘lips’.}%
\ProvidesPackage{lwarp-lips}

\NewDocumentCommand{\Lips}{}{\textellipsis}

\NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}

\let\lips\Lips
\let\olips\lips

\DeclareOption*{}
\DeclareOption{mla}{
\let\lips\BracketedLips
}
\ProcessOptions\relax

\newcommand \LPNobreakList {}
```

File 165 `lwarp-listings.sty`

§ 257 Package listings

(*Emulates or patches code by* Carsten Heinz, Brooks Moses, Jobst Hoffmann.)

Pkg `listings` **listings** is supported with some limitations. Text formatting is not yet supported.

for HTML output:

```
\begin{warpHTML}

\LWR@ProvidesPackagePass{listings}
```

Force flexible columns. Fixed columns inserts spaces in the PDF output.

```
\lst@column@flexible
```

Patches to embed listings inside `pre` tags:

```
\let\LWR@origlst@Init\lst@Init
\let\LWR@origlst@DeInit\lst@DeInit

\let\LWR@origlsthkEveryPar\lsthk@EveryPar

\renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{lstlisting}{lol}{#1}{#2}}
```

`\lstset` {⟨*options*⟩}

Use the **listings** `literate` option to replace HTML entities:

```
\def\lstset@#1{\endgroup%
%  \ifx\@empty#1%
%          \@empty%
%     \else%
        \setkeys{lst}{%
            #1%
            ,literate=%
            {<}{\HTMLentity{lt}}{4}%
            {>}{\HTMLentity{gt}}{4}%
            {\&}{\HTMLentity{amp}}{5}%
        }%
%     \fi%
}
```

`\lst@Init` {⟨*backslash-processing*⟩} Done at the start of a listing.

```
\renewcommand{\lst@Init}[1]{%
```

Perform the **listings** initialization:

```
\LWR@traceinfo{lst@Init}%
\renewcommand*{\@captype}{lstlisting}%
\let\lst@aboveskip\z@\let\lst@belowskip\z@%
\gdef\lst@boxpos{t}%
\let\lst@frame\@empty
    \let\lst@frametshape\@empty
    \let\lst@framershape\@empty
    \let\lst@framebshape\@empty
    \let\lst@framelshape\@empty
\lstframe@\lst@frameround ffff\relax%
\lst@multicols\@empty%
\LWR@origlst@Init{#1}\relax%
```

Avoids extra horizontal space:

```
\def\lst@framelr{}%
```

```
\LWR@traceinfo{finished origlst@Init}%
```

```
\lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```
\LWR@traceinfo{About to create verbatim.}%
\let\lsthk@EveryPar\relax%
\LWR@forcenewpage
\LWR@atbeginverbatim{2}{programlisting}%

\let\lsthk@EveryPar\LWR@origlsthkEveryPar%
\else%
```

Inline, so open a :

```
\ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
\fi%
}

```

`\lst@DeInit` Done at the end of a listing.

```
\renewcommand*{\lst@DeInit}{%
\lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```
\let\lsthk@EveryPar\relax%
\LWR@afterendverbatim{0}%
\let\lsthk@EveryPar\LWR@origlsthkEveryPar%
\else%
```

Inline, so create the closing :

```
\ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
\fi%
```

Final **listings** deinit:

```
\LWR@origlst@DeInit%
}
```

`\lst@MakeCaption` `{`⟨*t/b*⟩`}`

This is called BOTH at the top and at the bottom of each listing.

Patched for **lwarp**.

```
\def\lst@MakeCaption#1{%
\LWR@traceinfo{MAKING CAPTION at #1}%
  \lst@ifdisplaystyle
\LWR@traceinfo{making a listings display caption}%
    \ifx #1t%
```

```
        \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
                                  \expandafter\refstepcounter
        \fi {lstlisting}%
\LWR@traceinfo{About to assign label: !\lst@label!}%
%         \ifx\lst@label\@empty\else
% \label{\lst@label}\fi
\LWR@traceinfo{Finished assigning the label.}%
       \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
       \global\let\lst@name\lst@arg \global\let\lstname\lst@name
       \lst@ifnolol\else
           \ifx\lst@@caption\@empty
               \ifx\lst@caption\@empty
                   \ifx\lst@intname\@empty \else \def\lst@temp{ }%
                   \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for **lwarp**:

```
\LWR@traceinfo{addcontents lst@name: -\lst@name-}%
%                            \addcontentsline{lol}{lstlisting}{\lst@name}
                   \fi\fi
               \fi
           \else
```

This would have to be modified for **lwarp**:

```
\LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
                 \addcontentsline{lol}{lstlisting}%
{\protect\numberline{\thelstlisting}%
{\protect\ignorespaces \lst@@caption \protect\relax}}%
           \fi
         \fi
     \fi
    \ifx\lst@caption\@empty\else
\LWR@traceinfo{lst@caption not empty-}%
       \lst@IfSubstring #1\lst@captionpos
           {\begingroup
\LWR@traceinfo{at the selected position}%
```

These space and box commands are not needed for HTML output:

```
%             \let\@@vskip\vskip
%             \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
%             \def\lst@vskip{\nobreak\@@vskip\@tempskipa\nobreak}%
%             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
%             \ifx #1t\allowbreak \fi
            \ifx\lst@title\@empty
```

New **lwarp** code to create a caption:

```
                \lst@makecaption\fnum@lstlisting{\ignorespaces \lst@caption}
            \else
```

New **lwarp** code to create a title:

```
%                  \lst@maketitle\lst@title % (AS)
\LWR@traceinfo{Making title: \lst@title}%
\begin{BlockClass}{lstlistingtitle}% lwarp
\lst@maketitle\lst@title% lwarp
\end{BlockClass}% lwarp
             \fi
\LWR@traceinfo{About to assign label: !\lst@label!}%
        \ifx\lst@label\@empty\else
\leavevmode% gets rid of bad space factor error
\GetTitleStringExpand{\lst@caption}%
\edef\LWR@lntemp{\GetTitleStringResult}%
\edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
\label{\lst@label}\fi
\LWR@traceinfo{Finished assigning the label.}%
```

Not needed for **lwarp**:

```
%              \ifx #1b\allowbreak \fi
             \endgroup}{}%
    \fi
\LWR@traceinfo{end of making a listings display caption}%
  \else
\LWR@traceinfo{INLINE}%
  \fi
\LWR@traceinfo{DONE WITH CAPTION at #1}%
}
```

line numbers

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field `\VerbatimHTMLWidth` wide.

```
\lst@Key{numbers}{none}{%
    \let\lst@PlaceNumber\@empty
    \lstKV@SwitchCases{#1}%
    {none&\\%
     left&\def\lst@PlaceNumber{%
```

For now, **lwarp** places left line numbers inline. Ideally the entire line would be moved to the right, but conflicts with list indenting occurs.

```
%         \LWR@origllap{
            \LWR@orignormalfont%
            \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
%         }
    }\\%
     right&\def\lst@PlaceNumber{\LWR@origrlap{\LWR@orignormalfont
                \kern 6in \kern\lst@numbersep
                \lst@numberstyle{\thelstnumber}}}%
    }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}
```

```
\end{warpHTML}
```

File 166 `lwarp-longtable.sty`

§ 258 Package

longtable

(*Emulates or patches code by* David Carlisle.)

Pkg `longtable` **longtable** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{longtable}
```

⚠ Longtable `\endhead`, `\endfoot`, and `\endlastfoot` rows are not used for HTML, and these rows should be disabled. Use

```
\warpprintonly{row contents}
```

instead of

```
\begin{warpprint} ... \end{warpprint}
```

Doing so helps avoid "Misplaced `\noalign`." when using `\begin`{warpprint}.

Keep the `\endfirsthead` row, which is still relevent to HTML output.

⚠ `\kill` is ignored, place a `\kill` line inside

```
\begin{warpprint} ... \end{warpprint}
```

or place it inside `\warpingprintonly`.

⚠ lateximage **longtable** is not supported inside a `lateximage`.

See:
http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable

Env `longtable` * [⟨*horizalignment*⟩] {⟨*colspec*⟩} Emulates the `longtable` environment.

Per the **caption** package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options [c], [l], and [r] are thrown away.

```
\newenvironment{longtable*}[2][]{%
\LWR@floatbegin{table}%
\setcaptiontype{\LTcaptype}%
```

```
\caption@setoptions{longtable}%
\caption@setoptions{@longtable}%
\caption@LT@setup%
\booltrue{LWR@starredlongtable}%
\let\captionlistentry\LWR@LTcaptionlistentry%
\LWR@tabular{#2}
}
{\endLWR@tabular\LWR@floatend}

\newenvironment{longtable}[2][]{%
\LWR@floatbegin{table}%
\setcaptiontype{\LTcaptype}%
\caption@setoptions{longtable}%
\caption@setoptions{@longtable}%
\caption@LT@setup%
\refstepcounter{\LTcaptype}%
\let\captionlistentry\LWR@LTcaptionlistentry%
\LWR@tabular{#2}
}
{\endLWR@tabular\LWR@floatend}

```

Provided for compatibility, but ignored:

```
\newcounter{LTchunksize}
\def\endhead{\LWR@tabularendofline}% throws away options //[dim] and //*
\def\endfirsthead{\LWR@tabularendofline}
\def\endfoot{\LWR@tabularendofline}
\def\endlastfoot{\LWR@tabularendofline}
\newcommand\tabularnewline{\LWR@tabularendofline}
\newcommand{\setlongtables}{}% Obsolete command, does nothing.
\newlength{\LTleft}
\newlength{\LTright}
\newlength{\LTpre}
\newlength{\LTpost}
\newlength{\LTcapwidth}


\LetLtxMacro\LWR@origkill\kill
\renewcommand*{\kill}{\LWR@tabularendofline}
\appto\LWR@restoreorigformatting{%
\LetLtxMacro\kill\LWR@origkill%
}
```

File 167 `lwarp-lscape.sty`

§ 259 Package lscape

(*Emulates or patches code by* D. P. Carlisle.)

Pkg `lscape` **lscape** is emulated.

for HTML output: Discard all options for **lwarp-lscape**.

```
\LWR@ProvidesPackageDrop{lscape}

\newenvironment*{landscape}{}{}
```

File 168 `lwarp-ltablex.sty`

§ 260 Package ltablex

(*Emulates or patches code by* Anil K. Goel.)

Pkg `ltablex` **ltablex** is emulated by **lwarp**.

for HTML output: Relies on **tabularx**.

```
\RequirePackage{tabularx}

\LWR@ProvidesPackageDrop{ltablex}

\DeclareDocumentEnvironment{tabularx}{m o m}
{\longtable{#3}}
{\endlongtable}

\DeclareDocumentEnvironment{tabularx*}{m o m}
{\longtable{#3}}
{\endlongtable}

\newcommand*{\keepXColumns}{}
\newcommand*{\convertXColumns}{}
```

File 169 `lwarp-ltcaption.sty`

§ 261 Package ltcaption

(*Emulates or patches code by* Axel Sommerfeldt.)

Pkg `ltcaption` **ltcaption** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{ltcaption}
```

`\LTcaptype` is already defined by **lwarp**.

`longtable*` is already defined by **lwarp-longtable**.

```
\newlength{\LTcapskip}
\newlength{\LTcapleft}
\newlength{\LTcapright}
\newcommand*{\LTcapmarginsfalse}{}
```

File 170 `lwarp-ltxgrid.sty`

§ 262 Package ltxgrid

Pkg `ltxgrid` **ltxgrid** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{ltxgrid}
```

```
\newcommand*{\onecolumngrid}{}
\newcommand*{\twocolumngrid}{}
\newcommand*{\removestuff}{}
\newcommand*{\addstuff}[2]{}
\newcommand*{\replacestuff}[2]{}
```

File 171 `lwarp-ltxtable.sty`

§ 263 Package ltxtable

Pkg `ltxtable` **ltxtable** is emulated.

⚠ table numbering The print version does not seem to honor `longtable*` from the **caption** package, while **lwarp** does.

for HTML output:

```
\LWR@ProvidesPackageDrop{ltxtable}
```

`\LTXtable` {⟨*width*⟩} {⟨*file*⟩}

```
\newcommand*{\LTXtable}[2]{%
\input{#2}%
}
```

File 172 `lwarp-lua-check-hyphen.sty`

§ 264 Package lua-check-hyphen

Pkg `lua-check-hyphen` **lua-check-hyphen** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{lua-check-hyphen}
```

```
\newcommand*{\LuaCheckHyphen}[1]{}
```

File 173 `lwarp-luacolor.sty`

§ 265 Package luacolor

Pkg `luacolor` **luacolor** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{luacolor}
```

```
\newcommand{\luacolorProcessBox}[1]{}
```

File 174 `lwarp-luatodonotes.sty`

§ 266 Package luatodonotes

(*Emulates or patches code by* Fabian Lipp.)

Pkg `luatodonotes` **luatodonotes** is emulated.

The documentation for **todonotes** and **luatodonotes** have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

for HTML output:

```
\LWR@ProvidesPackagePass{luatodonotes}
```

Nullify options:

```
\@todonotes@additionalMarginEnabledfalse

\if@todonotes@disabled
\else

\newcommand{\ext@todo}{tdo}

\renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}


\let\LWRTODONOTES@orig@todototoc\todototoc

\renewcommand*{\todototoc}{%
\phantomsection%
\LWRTODONOTES@orig@todototoc%
}


\renewcommand{\@todonotes@drawMarginNoteWithLine}{%
\fcolorbox
    {\@todonotes@currentbordercolor}
    {\@todonotes@currentbackgroundcolor}
    {\arabic{@todonotes@numberoftodonotes}}
\marginpar{\@todonotes@drawMarginNote}
}

\renewcommand{\@todonotes@drawInlineNote}{%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \if@todonotes@authorgiven%
        {\@todonotes@author:\,}%
        \fi%
        \@todonotes@text%
    }%
}

\newcommand{\@todonotes@drawMarginNote}{%
    \if@todonotes@authorgiven%
        \@todonotes@author\par%
```

```
    \fi%
    \arabic{@todonotes@numberoftodonotes}: %
    \fcolorbox%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \@todonotes@sizecommand%
        \@todonotes@text %
    }%
}%

\renewcommand{\missingfigure}[2][]{%
\setkeys{todonotes}{#1}%
\addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentfigcolor}%
    {%
        \setlength{\fboxrule}{4pt}%
        \fcolorbox{red}{white}{Missing figure} \quad #2%
    }
}

\LetLtxMacro\LWRTODONOTES@orig@todocommon\@todocommon

\RenewDocumentCommand{\@todocommon}{m m}{%
\begingroup%
\renewcommand*{\phantomsection}{}%
\LWRTODONOTES@orig@todocommon{#1}{#2}%
\endgroup%
}

\renewcommand{\@todoarea}[3][]{%
    \@todonotes@areaselectedtrue%
    \@todocommon{#1}{#2}%
    \todonotes@textmark@highlight{#3}%
    \zref@label{@todonotes@\arabic{@todonotes@numberoftodonotes}@end}%
}%


\DeclareDocumentCommand{\todonotes@textmark@highlight}{m}{%
\InlineClass[background:\LWR@origpound{}B3FFB3]{highlight}{#1}%
}

\fi% \if@todonotes@disabled
```

File 175 `lwarp-magaz.sty`

§ 267 Package magaz

Pkg magaz **magaz** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{magaz}

\newcommand\FirstLine[1]{%
    \begingroup%
    \FirstLineFont{%
        \LWR@textcurrentcolor{%
            \LWR@textcurrentfont{%
                #1%
            }%
        }%
    }%
    \endgroup%
}
\providecommand\FirstLineFont{\scshape}
```

File 176 `lwarp-makeidx.sty`

§ 268 Package makeidx

(*Emulates or patches code by* LaTeX Project Team.)

Pkg makeidx **makeidx** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{makeidx}
```

`\@wrindex` is redefined `\AtBeginDocument` by the **lwarp** core.

`\printindex`

```
\preto\printindex{%
    \LWR@orignewpage%
    \LWR@startpars%
}
```

File 177 `lwarp-marginfit.sty`

§ 269 Package marginfit

Pkg `marginfit` **marginfit** is ignored.

for HTML output: Discard all options for **lwarp-marginfit**:

```
\LWR@ProvidesPackageDrop{marginfit}
```

File 178 `lwarp-marginfix.sty`

§ 270 Package marginfix

(*Emulates or patches code by* Stephen Hicks.)

Pkg `marginfix` Emulated.

for HTML output: Discard all options for **lwarp-marginfix**:

```
\LWR@ProvidesPackageDrop{marginfix}

\newcommand*{\marginskip}[1]{}
\newcommand*{\clearmargin}{}
\newcommand*{\softclearmargin}{}
\newcommand*{\extendmargin}[1]{}
\newcommand*{\mparshift}[1]{}
\newdimen\marginheightadjustment
\newdimen\marginposadjustment
\newcommand*{\blockmargin}[1][]{}
\newcommand*{\unblockmargin}[1][]{}
\newcommand*{\marginphantom}[2][]{}
```

File 179 `lwarp-marginnote.sty`

§ 271 Package marginnote

(*Emulates or patches code by* Markus Kohm.)

Pkg `marginnote` Emulated.

for HTML output: Discard all options for **lwarp-marginnote**:

```
\LWR@ProvidesPackageDrop{marginnote}

\NewDocumentCommand{\marginnote}{+o +m o}{\marginpar{#2}}

\newcommand*{\marginnoteleftadjust}{}
\newcommand*{\marginnoterightadjust}{}
\newcommand*{\marginnotetextwidth}{}
\let\marginnotetextwidth\textwidth
\newcommand*{\marginnotevadjust}{}
\newcommand*{\marginfont}{}
\newcommand*{\raggedleftmarginnote}{}
\newcommand*{\raggedrightmarginnote}{}
```

File 180 `lwarp-mcaption.sty`

§ 272 Package mcaption

(*Emulates or patches code by* Stephan Hennig.)

Pkg `mcaption` **mcaption** is nullified.

for HTML output: Discard all options for **lwarp-mcaption**:

```
\LWR@ProvidesPackageDrop{mcaption}

\newenvironment{margincap}{}{}
\newcommand*{\margincapalign}{}
\newlength{\margincapsep}
```

File 181 `lwarp-mdframed.sty`

§ 273 Package mdframed

(*Emulates or patches code by* Marco Daniel, Elke Schubert.)

Pkg `mdframed` **mdframed** is loaded with options forced to `framemethod=none`.

§ 273.1 Limitations

support Most basic functionality is supported, including frame background colors and single-border colors and thickness, title and subtitle background colors and borders and thickness, border radius, and shadow. CSS classes are created for **mdframed** environments and frame titles.

⚠ loading When used, **lwarp** loads **mdframed** in HTML with `framemethod=none`.

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the **mdframed** source). Since **lwarp** does not support `\bfseries` and friends, only one font selection may be made at a time.

theoremtitlefont `theoremtitlefont` is not supported, since the following text is not in braces in the **mdframed** source.

footnotes Footnotes are currently placed at the bottom of the HTML page.

ignored options `userdefinedwidth` and `align` are currently ignored.

CSS classes Environments created or encapsulated by **mdframed** are enclosed in a `<div>` of class `md<environmentname>`, or `mdframed` otherwise.

Frame titles are placed into a `<span>` of class `mdframedtitle`. Subtitles are in a `<span>` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

§ 273.2 Package loading

for HTML output:

```
\RequirePackage{xcolor}% for \convertcolorspec

\LWR@ProvidesPackageDrop{mdframed}
```

amsthm must be loaded before **mdframed**

```
\LWR@origRequirePackage{amsthm}
```

Do not require Ti*k*z or pstricks:

```
\LWR@origRequirePackage[framemethod=none]{mdframed}
```

§ 273.3 Patches

Patch to remove PDF formatting and add HTML tags:

```
\AtBeginDocument{
\def\mdf@trivlist#1{%
  \edef\mdf@temp{%
%     \topsep=\the\topsep\relax%
%     \partopsep=\the\partopsep\relax%
%     \parsep=\the\parsep\relax%
  }%
%   \setlength{\topsep}{#1}%
%   \topskip\z@%
%   \partopsep\z@%
%   \parsep\z@%
%   \@nmbrlistfalse%
%   \@trivlist%
%   \labelwidth\z@%
%   \leftmargin\z@%
%   \itemindent\z@%
  \let\@itemlabel\@empty%
  \def\makelabel##1{##1}%
%   \item\relax\mdf@temp\relax%
}

\renewcommand*{\endmdf@trivlist}{%
\LWR@traceinfo{endmdf@trivlist}%
% \endtrivlist%
\LWR@listend%
}
}% AtBeginDocument
```

§ 273.4 Initial setup

To handle CSS and paragraphs, patch code at start and end of environment and contents. `\LWR@print@raggedright` helps avoid hyphenation.

```
\mdfsetup{
startcode={\LWR@mdframedstart\LWR@print@raggedright},
endcode={\LWR@mdframedend},
startinnercode={\LWR@startpars\LWR@print@raggedright},
endinnercode={\LWR@stoppars},
}
```

§ 273.5 Color and length HTML conversion

\LWR@mdfprintcolor {⟨*mdfcolorkey*⟩}

Given the **mdframed** key, print the color.

```
\newcommand*{\LWR@mdfprintcolor}[1]{%
\convertcolorspec{named}{\@nameuse{mdf@#1}}{HTML}\LWR@tempcolor%
\LWR@origpound\LWR@tempcolor
}
```

\LWR@mdfprintlength {⟨*mdflengthkey*⟩}

Given the **mdframed** key, print the length.

```
\newcommand*{\LWR@mdfprintlength}[1]{%
\LWR@printlength{\@nameuse{mdf@#1@length}}
}
```

§ 273.6 Environment encapsulation

\LWR@mdframedstart Actions before an mdframe starts.

Encapsulate a frame inside a `<div>` of the desired `class`.

```
\newcommand*{\LWR@mdframedstart}{%
\LWR@traceinfo{LWR@mdframedstart start}%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
\LWR@stoppars%
```

Open a `<div>` and with custom `class` and custom `style`:

```
\LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
style=" \LWR@orignewline
```

Convert and print the background color:

```
background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
border: \LWR@mdfprintlength{linewidth} solid
\LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
\ifbool{mdf@shadow}{%
    box-shadow:
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintlength{shadowsize}
    \LWR@mdfprintcolor{shadowcolor} ;
```

```
}
{box-shadow: none ;}
\LWR@orignewline

"}
% \LWR@htmldivclass{\LWR@mdthisenv}
```

mdframed environment may not work with the HTML versions of the following, so restore them to their originals while inside mdframed:

```
\LWR@select@print@hspace%
\renewcommand*{\rule}{\LWR@print@rule}
\LetLtxMacro\makebox\LWR@print@makebox%

\LWR@startpars%
\LWR@traceinfo{LWR@mdframedstart done}%
}
```

\LWR@mdframedend Actions after an mdframe ends.

After closing the , globally restore to the default environment type:

```
\newcommand*{\LWR@mdframedend}{
\LWR@traceinfo{LWR@mdframedend start}%
```

Close the custom :

```
\LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
\gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
\LWR@startpars%
\LWR@traceinfo{LWR@mdframedend done}%
}
```

§ 273.7 Mdframed environment

```
\renewenvironment{mdframed}[1][]{%
 \color@begingroup%
   \mdfsetup{userdefinedwidth=\linewidth,#1}%
   \mdf@startcode%
   \mdf@preenvsetting%
   \ifdefempty{\mdf@firstframetitle}{}%
           {\let\mdf@frametitlesave\mdf@frametitle%
            \let\mdf@frametitle\mdf@firstframetitle%
           }%
   \ifvmode\nointerlineskip\fi%
        \ifdefempty{\mdf@frametitle}{}%
            {\mdfframedtitleenv{\mdf@frametitle}%
%             \mdf@@frametitle@use%
```

```
            }%
  \mdf@trivlist{\mdf@skipabove@length}%%
  \mdf@settings%
%    \mdf@lrbox{\mdf@splitbox@one}%
%    \mdf@startinnercode%
 }%
 {%
%    \mdf@@ignorelastdescenders%
  \par%
%     \unskip\ifvmode\nointerlineskip\hrule \@height\z@ \@width\hsize\fi%%
  \ifmdf@footnoteinside%
     \def\mdf@reserveda{%
       \mdf@footnoteoutput%
%        \mdf@endinnercode%
%        \endmdf@lrbox%
%        \ifdefempty{\mdf@frametitle}{}%
%            {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
%        \detected@mdf@put@frame
     }%
  \else%
     \def\mdf@reserveda{%
%         \mdf@endinnercode%
%         \endmdf@lrbox%
%         \ifdefempty{\mdf@frametitle}{}%
%             {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
%         \detected@mdf@put@frame%
       \mdf@footnoteoutput%
       }%
  \fi%
  \mdf@reserveda%
 \aftergroup\endmdf@trivlist%
\color@endgroup%
\mdf@endcode%
}
```

\mdf@footnoteoutput

```
\renewrobustcmd*\mdf@footnoteoutput{%
    \LWR@printpendingmpfootnotes%
}
```

§ 273.8 Titles and subtitles

\mdfframedtitleenv {⟨*title*⟩}

Encapsulation of the original which places the title inside a of class mdframedtitle:

```
\LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv

\newlength{\LWR@titleroundcorner}

```

```
\renewrobustcmd\mdfframedtitleenv[1]{%
\LWR@traceinfo{LWR@mdframedtitleenv start}%
% \LWR@origmdfframedtitleenv{%
```

Open a with a custom class and custom style:

```
\LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
style=" \LWR@orignewline
```

Convert and print the title background color:

```
background:
\LWR@mdfprintcolor{frametitlebackgroundcolor}
; \LWR@orignewline
```

Convert and print the title rule:

```
\ifbool{mdf@frametitlerule}{%
    border-bottom:
    \LWR@mdfprintlength{frametitlerulewidth}
    solid
    \LWR@mdfprintcolor{frametitlerulecolor}
    ; \LWR@orignewline
}{}%
```

Finish the custom style and the opening span tag:

```
" \LWR@orignewline
}% span
```

Restrict paragraph tags inside a span:

```
\begin{LWR@nestspan}%
```

Print the title inside the span:

```
#1%
```

Closee the span and unnest the paragraph tag restriction:

```
\LWR@htmltagc{/span}%
\end{LWR@nestspan}%
% }
\LWR@traceinfo{LWR@mdframedtitleenv end}%
}
```

`\LWR@mdfsubtitlecommon` {⟨*sub -or- subsub*⟩} [⟨*options*⟩] {⟨*title*⟩}

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a of class mdframedsubtitle:

```
\NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
{% the following empty line is required

\LWR@traceinfo{LWR@mdframedsubtitlecommon start}%
```

Special handling for mdframed: Subtitles have \pars around them, so temporarily disable them here.

```
\let\par\LWR@origpar%
```

Open a with a custom class and custom style:

```
\LWR@htmltagc{span class="mdframed#1title"
style=" \LWR@orignewline
```

Convert and print the background color:

```
background:
\LWR@mdfprintcolor{#1titlebackgroundcolor}
; \LWR@orignewline
```

Convert and print the above line:

```
\ifbool{mdf@#1titleaboveline}{%
    border-top:
    \LWR@mdfprintlength{#1titleabovelinewidth}
    solid
    \LWR@mdfprintcolor{#1titleabovelinecolor}
    ; \LWR@orignewline
}{}%
```

Convert and print the below line:

```
\ifbool{mdf@#1titlebelowline}{%
    border-bottom:
    \LWR@mdfprintlength{#1titlebelowlinewidth}
    solid
    \LWR@mdfprintcolor{#1titlebelowlinecolor}
    ; \LWR@orignewline
}{}%
```

Finish the custom style and the opening span tag:

```
"}% span
```

Restrict paragraph tags inside a span:

```
\begin{LWR@nestspan}%
```

Perform the original subtitle action:

```
\IfNoValueTF{#2}
{\@nameuse{LWR@origmdf#1title}{#3}}%
{\@nameuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```
\LWR@htmltagc{/span}% the following empty line is required
\end{LWR@nestspan}% must follow the /span or an extra <p> appears

\LWR@traceinfo{LWR@mdframedsubtitlecommon end}%
}
```

\LWR@mdfsubtitle [⟨*options*⟩] {⟨*title*⟩}

```
\newcommand*{\LWR@mdfsubtitle}{%
\LWR@mdfsubtitlecommon{sub}%
}
\let\mdfsubtitle\LWR@mdfsubtitle
```

\LWR@mdfsubsubtitle [⟨*options*⟩] {⟨*title*⟩}

```
\newcommand*{\LWR@mdfsubsubtitle}{%
\LWR@mdfsubtitlecommon{subsub}%
}
\let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

§ 273.9 New environments

\LWR@mdthisenv Stores the environment of the frame about to be created:

```
\newcommand*{\LWR@mdthisenv}{mdframed}
```

\newmdenv [⟨*options*⟩] {⟨*env-name*⟩}

Modified from the original to remember the environment.

```
\renewrobustcmd*\newmdenv[2][]{%
\newenvironment{#2}%
{%
\mdfsetup{#1}%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}%
}
{\end{mdframed}}%
}
```

\surroundwithmdframed [⟨*options*⟩] {⟨*environment*⟩}

Modified from the original to remember the environment.

```
\renewrobustcmd*{\surroundwithmdframed}[2][]{%
\BeforeBeginEnvironment{#2}{%
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}[#1]}%
\AfterEndEnvironment{#2}{\end{mdframed}}%
}
```

\mdtheorem [⟨*mdframed-options*⟩] {⟨*envname*⟩} [⟨*numberedlike*⟩] {⟨*caption*⟩} [⟨*within*⟩]

Modified from the original to remember the environment.

```
\DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }%
```

```
{\ifcsdef{#2}%
  {\mdf@PackageWarning{Environment #2 already exits\MessageBreak}}%
  {%
   \IfNoValueTF {#3}%
    {%#3 not given -- number relationship
     \IfNoValueTF {#5}%
       {%#3+#5 not given
       \@definecounter{#2}%
       \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
       \newenvironment{#2}[1][]{%
         \refstepcounter{#2}%
         \ifstrempty{##1}%
           {\let\@temptitle\relax}%
           {%
            \def\@temptitle{\mdf@theoremseparator%
                            \mdf@theoremspace%
                            \mdf@theoremtitlefont%
                            ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }%
         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                         \@temptitle}]}%
         {\end{mdframed}}%
       \newenvironment{#2*}[1][]{%
         \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{:\ ##1}}%
         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
         {\end{mdframed}}%
       }%
       {%#5 given -- reset counter
       \@definecounter{#2}\@newctr{#2}[#5]%
       \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
       \expandafter\xdef\csname the#2\endcsname{%
              \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
                 \@thmcounter{#2}}%
       \newenvironment{#2}[1][]{%
         \refstepcounter{#2}%
         \ifstrempty{##1}%
           {\let\@temptitle\relax}%
           {%
            \def\@temptitle{\mdf@theoremseparator%
                            \mdf@theoremspace%
                            \mdf@theoremtitlefont%
                            ##1}%
            \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
            }
         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                         \@temptitle}]}%
         {\end{mdframed}}%
       \newenvironment{#2*}[1][]{%
         \ifstrempty{##1}%
```

```
            {\let\@temptitle\relax}%
            {%
             \def\@temptitle{\mdf@theoremseparator%
                             \mdf@theoremspace%
                             \mdf@theoremtitlefont%
                             ##1}%
             \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
             }%
          \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
          {\end{mdframed}}%
        }%
     }%
     {%#3 given -- number relationship
        \global\@namedef{the#2}{\@nameuse{the#3}}%
        \newenvironment{#2}[1][]{%
          \refstepcounter{#3}%
          \ifstrempty{##1}%
            {\let\@temptitle\relax}%
            {%
             \def\@temptitle{\mdf@theoremseparator%
                             \mdf@theoremspace%
                             \mdf@theoremtitlefont%
                             ##1}%
             \mdf@thm@caption{#2}{{#4}{\csname the#2\endcsname}{##1}}%
             }
          \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
                                          \@temptitle}]}%
          {\end{mdframed}}%
        \newenvironment{#2*}[1][]{%
          \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{:\ ##1}}%
          \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]}%
          {\end{mdframed}}%
     }%
    \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% lwarp
    \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}% lwarp
   }%
}
```

\newmdtheoremenv `[⟨mdframed-options⟩] {⟨envname⟩} [⟨numberedlike⟩] {⟨caption⟩} [⟨within⟩]`

Modified from the original to remember the environment.

```
\DeclareDocumentCommand\newmdtheoremenv{O{} m o m o }{%
 \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
    {\newtheorem{#2}{#4}}%
    {%
     \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
     \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
    }%
\BeforeBeginEnvironment{#2}{%
```

```
\renewcommand*{\LWR@mdthisenv}{md#2}%
\begin{mdframed}[#1]}%
\AfterEndEnvironment{#2}{%
\end{mdframed}}%
}
```

File 182 `lwarp-memhfixc.sty`

§ 274 Package memhfixc

Pkg `memhfixc` **memhfixc** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{memhfixc}
```

File 183 `lwarp-metalogo.sty`

§ 275 Package metalogo

(*Emulates or patches code by* Andrew Gilbert Moschou.)

Pkg `metalogo` **metalogo** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{metalogo}

\newcommand\setlogokern[2]{}
\newcommand\setlogodrop[2][XeTeX]{}
\newcommand\setLaTeXa[1]{}
\newcommand\setLaTeXee[1]{}
\newcommand\seteverylogo[1]{}
\newcommand\everylogo[1]{}
```

File 184 `lwarp-mhchem.sty`

§ 276 Package mhchem

(*Emulates or patches code by* Martin Hensel.)

Pkg `mhchem` **mhchem** is patched for use by **lwarp**.

mhchem expressions are converted to SVG math. Inline expressions use hashed filenames to allow reuse, and assume that any **mhchem** options are global.

⚠ MathJax and **mhchem**

The MathJax **mhchem** extension is not yet used. If MathJax is used for math in the rest of the document, **lwarp** converts standalone **mhchem** expressions into SVG math images, but expressions inside math must be placed between `\displaymathother` and `\displaymathnormal`:

```
\displaymathother
\[ \ce{ ... } \]     ...    $ \ce { ... } $
\displaymathnormal
```

⚠ nested math

When producing HTML output, **lwarp** does not support the use of nested dollar signs in **mhchem** expressions.

For some examples from the **mhchem** manual, change as follows:

```
$\ce{NaOH(aq,$\infty$)}$            % old
$\ce{NaOH(aq,\infty)}$              % new

$\ce{Fe(CN)_{$\frac{6}{2}$}$}$      % old
$\ce{Fe(CN)_{\frac{6}{2}}}$         % new

$\ce{NO_$x$}$                       % old
$\ce{NO_x}$                         % new

$\ce{NO_${x}$}$                     % old
$\ce{NO_{x}}$                       % new

$\ce{$cis${-}[PtCl2(NH3)2]}$        % old
$\ce{\mathit{cis}{-}[PtCl2(NH3)2]}$ % new
```

for HTML output:

```
\LWR@ProvidesPackagePass{mhchem}
```

The original definition of `\ce`:

```
\LetLtxMacro\LWR@mhchem@origce\ce
```

The new definition, called from the new `\ce` after math shift is set. The starred `lateximage` uses a hashed filename for the SVG image. The `alt` tag is set to the **mhchem** expression.

```
\newcommand{\LWR@mhchem@HTML@ce}[1]{%
\begin{lateximage}*[\textbackslash{}ce\{\LWR@HTMLsanitize{#1}\}]%
\LWR@mhchem@origce{#1}%
\end{lateximage}%
\endgroup%
\addtocounter{LWR@mhchem@cedepth}{-1}%
}
```

Only set math shift if outer depth:

```
\newcounter{LWR@mhchem@cedepth}
\setcounter{LWR@mhchem@cedepth}{0}
```

The new `\ce`. Sets math shift then continues.

```
\renewcommand{\ce}{%
\begingroup%
\ifnumequal{\value{LWR@mhchem@cedepth}}{0}{%
    \catcode`\$=3% math shift
}{}%
\addtocounter{LWR@mhchem@cedepth}{1}%
\LWR@mhchem@HTML@ce%
}
```

The original definition of `\cesplit`:

```
\LetLtxMacro\LWR@mhchem@origcesplit\cesplit
```

The new definition, called from the new `\cesplit` after math shift is set. The starred `lateximage` uses a hashed filename for the SVG image. The `alt` tag is set to the **mhchem** expression.

```
\newcommand*{\LWR@mhchem@HTML@cesplit}[2]
{%
\begin{lateximage}*[\textbackslash{}cesplit\{\LWR@HTMLsanitize{#2}\}]%
\LWR@mhchem@origcesplit{#1}{#2}%
\end{lateximage}%
\endgroup%
}
```

Only set math shift if outer depth:

```
\newcounter{LWR@mhchem@cesplitdepth}
\setcounter{LWR@mhchem@cesplitdepth}{0}
```

The new `\cesplit`. Sets math shift then continues.

```
\renewcommand{\cesplit}{%
\begingroup%
\ifnumequal{\value{LWR@mhchem@cesplitdepth}}{0}{%
    \catcode`\$=3% math shift
}{}%
\addtocounter{LWR@mhchem@cesplitdepth}{1}%
\LWR@mhchem@HTML@cesplit%
}
```

Resore originals inside a `lateximage`:

```
\appto\LWR@restoreorigformatting{%
\LetLtxMacro\ce\LWR@mhchem@origce%
\LetLtxMacro\cesplit\LWR@mhchem@origcesplit%
}
```

File 185 `lwarp-microtype.sty`

§ 277 Package

microtype

(*Emulates or patches code by* R Schlicht.)

Pkg `microtype` **microtype** is pre-loaded by **lwarp**. All user options and macros are ignored and disabled.

for HTML output: Discard all options for **lwarp-microtype**:

```
\LWR@ProvidesPackageDrop{microtype}

\DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
\DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
\DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
\DeclareDocumentCommand{\SetProtrusion}{o m m}{}
\DeclareDocumentCommand{\SetExpansion}{o m m}{}
\DeclareDocumentCommand{\SetTracking}{o m m}{}
\DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
\DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
\DeclareDocumentCommand{\DisableLigatures}{o m}{}
\DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
\DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
\DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
\DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
\DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
\DeclareDocumentCommand{\microtypesetup}{m}{}
\DeclareDocumentCommand{\microtypecontext}{m}{}
\DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
\@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
\DeclareDocumentCommand{\lsstyle}{}{}
\DeclareDocumentCommand{\textls}{o +m}{}
\DeclareDocumentCommand{\lslig}{m}{#1}
}
\def\DeclareMicrotypeSet#1#{\@gobbletwo}
\def\DeclareMicrotypeVariants#1#{\@gobble}
\@onlypreamble\DeclareMicrotypeSet
\@onlypreamble\UseMicrotypeSet
\@onlypreamble\DeclareMicrotypeSetDefault
```

```
\@onlypreamble\DisableLigatures
\@onlypreamble\DeclareMicrotypeVariants
\@onlypreamble\DeclareMicrotypeBabelHook
```

File 186 `lwarp-midfloat.sty`

§ 278 Package midfloat

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `midfloat` **midfloat** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{midfloat}
```

```
\newenvironment{strip}[1][]{}{}
\newskip\stripsep
```

File 187 `lwarp-midpage.sty`

§ 279 Package midpage

Pkg `midpage` **midpage** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{midpage}
```

```
\newenvironment{midpage}
{\begin{BlockClass}[\LWR@print@mbox{margin-top:6ex} ; \LWR@print@mbox{margin-bottom:6ex}]{midpa
{\end{BlockClass}}
```

File 188 `lwarp-morefloats.sty`

§ 280 Package morefloats

Pkg `morefloats` **morefloats** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{morefloats}
```

File 189 `lwarp-moreverb.sty`

§ 281 Package moreverb

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `moreverb` **moreverb** is supported with some patches.

for HTML output:

```
\begin{warpHTML}

\LWR@ProvidesPackagePass{moreverb}

\BeforeBeginEnvironment{verbatimtab}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{Verbatim}%
}
\AfterEndEnvironment{verbatimtab}{%
\LWR@afterendverbatim{1}%
}


\LetLtxMacro\LWRMV@orig@verbatimtabinput\@verbatimtabinput

\renewcommand{\@verbatimtabinput}[2][]{%
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{Verbatim}%
\LWRMV@orig@verbatimtabinput[#1]{#2}%
\LWR@afterendverbatim{1}%
}

\BeforeBeginEnvironment{listing}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{programlisting}%
}

\AfterEndEnvironment{listing}{%
\LWR@afterendverbatim{1}%
}

\BeforeBeginEnvironment{listingcont}{%
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{programlisting}%
}

\AfterEndEnvironment{listingcont}{%
```

```
\LWR@afterendverbatim{1}%
}

\LetLtxMacro\LWRMV@@listinginput\@listinginput

\renewcommand{\@listinginput}[3][]{
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{programlisting}%
\LWRMV@@listinginput[#1]{#2}{#3}%
\LWR@afterendverbatim{1}%
}


\renewenvironment*{boxedverbatim}
{
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{boxedverbatim}%
\verbatim%
}
{
\endverbatim%
\LWR@afterendverbatim{1}%
}

\end{warpHTML}
```

File 190 `lwarp-morewrites.sty`

§ 282 Package morewrites

Pkg `morewrites` Error if **morewrites** is loaded after **lwarp**.

Discard all options for **lwarp-morewrites**:

for HTML output:

```
\LWR@ProvidesPackageDrop{morewrites}

\LWR@loadbefore{morewrites}
```

File 191 `lwarp-mparhack.sty`

§ 283 Package mparhack

Pkg `mparhack` Ignored.

for HTML output: Discard all options for **lwarp-mparhack**:

```
\LWR@ProvidesPackageDrop{mparhack}
```

File 192 `lwarp-multicol.sty`

§ 284 Package multicol

(*Emulates or patches code by* Frank Mittelbach.)

Pkg `multicol` **multicol** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{multicol}[2015/09/13]
```

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a `<div>` of class `multicolsheading`.

The content is placed inside a `<div>` of class `multicols`.

```
\begin{warpHTML}
```

Env `multicols` * {⟨*numcols*⟩} [⟨*heading*⟩]

```
\NewDocumentEnvironment{multicols}{s m o}
```

HTML `<div>` class to contain everything:

```
{
\LWR@forcenewpage
\BlockClass{multicols}
```

Optional HTML `<div>` class for the heading:

```
\IfValueT{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}}
```

When done with the environment, close the `<div>`:

```
{\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
\newcommand*{\columnbreak}{}
\newcommand*{\RLmulticolcolumns}{}
\newcommand*{\LRmulticolcolumns}{}

\newlength{\premulticols}
\newlength{\postmulticols}
\newlength{\multicolsep}
```

```
\newlength{\multicolbaselineskip}
\newlength{\multicoltolerance}
\newlength{\multicolpretolerance}
\newcommand*{\columnseprulecolor}{\normalcolor}
\newcounter{columnbadness}
\newcounter{finalcolumnbadness}
\newcounter{collectmore}
\newcounter{unbalance}
\newlength{\multicolovershoot}
\newlength{\multicolundershoot}

\end{warpHTML}
```

File 193 `lwarp-multirow.sty`

§ 285 Package

multirow

(*Emulates or patches code by* Piet van Oostrum, Øystein Bache, Jerry Leichter.)

Pkg `multirow`

multirow is emulated during HTML output, and used as-is while inside a `lateximage`.

In a `lateximage`, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 68.23 for the print-mode versions.

for HTML output:

Remove the placeholder macro which was used if **multirow** was not loaded:

```
\LetLtxMacro\multirow\relax

\LWR@ProvidesPackagePass{multirow}
```

`\LWR@multirowborder`

Set to `left` or `right` to create a thick border for the cell, for use by **bigdelim**:

```
\newcommand{\LWR@multirowborder}{}
```

§ 285.1 Multirow

\multirow [⟨*vpos*⟩] {⟨*numrows*⟩} [⟨*bigstruts*⟩] {⟨*width*⟩} [⟨*fixup*⟩] {⟨*text*⟩}

```
\NewDocumentCommand{\LWR@HTML@multirow}{O{c} m o m o +m}%
{%
\LWR@traceinfo{*** LWR@HTML@multirow #1 #2 #4}%
\LWR@maybenewtablerow%
\LWR@tabularleftedge%
```

Print the start of a new table data cell:

```
\LWR@htmltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
\IfValueT{#1}{%
\ifstrequal{#1}{b}{style="\LWR@print@mbox{vertical-align:bottom}" }{}%
\ifstrequal{#1}{t}{style="\LWR@print@mbox{vertical-align:top}" }{}%
}%
```

The left/right border, if given:

```
\ifdefvoid{\LWR@multirowborder}{}{%
style="\LWR@print@mbox{border-\LWR@multirowborder:} 2px dotted black ; %
\LWR@print@mbox{padding-\LWR@multirowborder:} 2px" %
}%
```

A class adds the column spec and the rule:

```
class="td%
```

Append this column's spec:

```
\LWR@getexparray{LWR@tablecolspec}{\arabic{LWR@tableLaTeXcolindex}}%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add the vertical bar class.

```
\LWR@addcmidruletrim%
\LWR@addleftmostbartag%
\LWR@printbartag{\arabic{LWR@tableLaTeXcolindex}}%
"%


\LWR@tdstartstyles%
\LWR@addcmidrulewidth%
\LWR@addcdashline%
```

```
\LWR@addtabularrulecolors%
\LWR@tdendstyles%
}%
```

The column's < spec:

```
\LWR@getexparray{LWR@colbeforespec}{\arabic{LWR@tableLaTeXcolindex}}%
```

While printing the text, redefine \\ to generate a new line

```
\begingroup\LetLtxMacro{\\}{\LWR@endofline}#6\endgroup%
\LWR@stoppars%
\global\boolfalse{LWR@intabularmetadata}%
\renewcommand{\LWR@multirowborder}{}%
\LWR@traceinfo{*** LWR@HTML@multirow done}%
}%

\LWR@formatted{multirow}
```

§ 285.2 Combined multicolumn and multirow

⚠ \multicolumn & \multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents.

⚠ skipped cells

⚠ empty cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

```
... &  \multicolumnrow{2}{c}[c]{3}[0]{1in}[0pt]{Text}  & ...
... &  \mcolrowcell           &           \mcolrowcell  & ...
... &  \mcolrowcell           &           \mcolrowcell  & ...
```

vposn

Note that recent versions of **multirow** include a new optional vposn argument.

\multicolumnrow {⟨*1:cols*⟩} {⟨*2:halign*⟩} [⟨*3:vpos*⟩] {⟨*4:numrows*⟩} [⟨*5:bigstruts*⟩] {⟨*6:width*⟩} [⟨*7:fixup*⟩] {⟨*8:text*⟩}

\@ifpackageloaded{multirow} determines if v2.0 or later of **multirow** was used, which included the \ProvidesPackage macro.

The HTML version follows.

\AtBeginDocument because the print version had to see if **multirow** was loaded before determining how to define \LWR@print@multicolumnrow.

```
\AtBeginDocument{

\NewExpandableDocumentCommand{\LWR@HTML@multicolumnrow}{m m O{} m O{} m O{} +m}{%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
\LWR@tabularhtmlcolumns{\arabic{LWR@tableLaTeXcolindex}}{#1}
```

Create the multicolumn/multirow tag:

```
\begingroup%
\LetLtxMacro{\\}{\LWR@endofline}%
\LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
\endgroup%
```

Move to the next LaTeX column:

```
\addtocounter{LWR@tableLaTeXcolindex}{#1}%
\addtocounter{LWR@tableLaTeXcolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
\global\booltrue{LWR@skipatbang}%
}

\LWR@expandableformatted{multicolumnrow}

}% \AtBeginDocument
```

File 194 `lwarp-multitoc.sty`

§ 286 Package

multitoc

Pkg multitoc **multitoc** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{multitoc}

\newcommand{\multicolumntoc}{2}
\newcommand{\multicolumnlot}{2}
\newcommand{\multicolumnlof}{2}
\newcommand*{\immediateaddtocontents}[2]{}
```

File 195 `lwarp-nameref.sty`

§ 287 Package nameref

Pkg `nameref` **nameref** is emulated by **lwarp**.

for HTML output: Discard all options for **lwarp-nameref**:

```
\typeout{Using the lwarp html version of package ‘nameref’, discarding options.}
\typeout{    Are not using ProvidesPackage, so that other packages}
\typeout{    do not attempt to patch lwarp’s version of ‘nameref’.}
\DeclareOption*{}
\ProcessOptions\relax
```

File 196 `lwarp-natbib.sty`

§ 288 Package natbib

(*Emulates or patches code by* Patrick W. Daly.)

Pkg `natbib` **natbib** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{natbib}
```

Replace math < and > with `\textless` and `\textgreater`:

A macro to compare:

```
\newcommand{\LWRNB@NAT@open}{$<$}
```

To patch `\NAT@open` and `\NAT@close`

```
\newcommand{\LWRNB@patchnatbibopenclose}{
\ifdefstrequal{\NAT@open}{\LWRNB@NAT@open}
{
    \renewcommand{\NAT@open}{\textless}
    \renewcommand{\NAT@close}{\textgreater}
}{}
}
```

Do it now in case `angle` was selected as an option:

```
\LWRNB@patchnatbibopenclose
```

Also patch \setcitestyle to patch after settings are made:

```
\let\LWRNB@origsetcitestyle\setcitestyle

\renewcommand{\setcitestyle}[1]{%
\LWRNB@origsetcitestyle{#1}%
\LWRNB@patchnatbibopenclose%
}
```

File 197 lwarp-nccfancyhdr.sty

§ 289 Package

nccfancyhdr

(*Emulates or patches code by* Alexander I. Rozhenko.)

Pkg nccfancyhdr **nccfancyhdr** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{nccfancyhdr}

\newcommand*{\headrulewidth}{}
\newcommand*{\footrulewidth}{}
\newcommand{\headstrutheight}{}
\newcommand{\footstrutheight}{}
\newcommand*{\headrule}{}
\newcommand*{\footrule}{}

\newdimen\headwidth
\newcommand*{\extendedheaders}{}
\newcommand*{\normalheaders}{}

\newcommand*{\fancyhead}[2][]{}
\newcommand*{\fancyfoot}[2][]{}
\newcommand*{\fancyhf}[2][]{}
\newcommand*{\fancypagestyle}[2]{}
\newcommand*{\lhead}[2][]{}
\newcommand*{\chead}[2][]{}
\newcommand*{\rhead}[2][]{}
\newcommand*{\lfoot}[2][]{}
\newcommand*{\cfoot}[2][]{}
\newcommand*{\rfoot}[2][]{}

\newcommand{\nouppercase}[1]{#1}

\NewDocumentCommand{\fancycenter}{o o m m m}{}

\NewDocumentCommand{\newpagestyle}{m o m}{}

```

```
\newcommand*{\iffloatpage}[2]{#2}
\newcommand*{\ifftopfloat}[2]{#2}
\newcommand*{\iffbotfloat}[2]{#2}
```

File 198 `lwarp-needspace.sty`

§ 290 Package needspace

(*Emulates or patches code by* Peter Wilson.)

Pkg `needspace` **needspace** is not used during HTML conversion.

for HTML output: Discard all options for **lwarp-needspace**:

```
\LWR@ProvidesPackageDrop{needspace}

\DeclareDocumentCommand{\needspace}{m}{}
\DeclareDocumentCommand{\Needspace}{s m}{}
```

File 199 `lwarp-newclude.sty`

§ 291 Package newclude

Pkg `newclude` Error if **newclude** is loaded after **lwarp**.

Discard all options for **lwarp-newclude**:

for HTML output:

```
\LWR@ProvidesPackageDrop{newclude}

\LWR@loadbefore{newclude}
```

File 200 `lwarp-newunicodechar.sty`

§ 292 Package newunicodechar

Pkg `newunicodechar` Error if **newunicodechar** is loaded after **lwarp**.

Discard all options for **lwarp-newunicodechar**:

for HTML output:

```
\LWR@ProvidesPackageDrop{newunicodechar}

\LWR@loadbefore{newunicodechar}
```

File 201 `lwarp-nextpage.sty`

§ 293 Package nextpage

(*Emulates or patches code by* Peter Wilson.)

Pkg `nextpage` **nextpage** is nullified.

for HTML output: Discard all options for **lwarp-nextpage**.

```
\LWR@ProvidesPackageDrop{nextpage}

\DeclareDocumentCommand{\cleartoevenpage}{o}{}
\DeclareDocumentCommand{\movetoevenpage}{o}{}
\DeclareDocumentCommand{\cleartooddpage}{o}{}
\DeclareDocumentCommand{\movetooddpage}{o}{}
```

File 202 `lwarp-nicefrac.sty`

§ 294 Package nicefrac

(*Emulates or patches code by* Axel Reichert.)

Pkg `nicefrac` **nicefrac** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{nicefrac}[1998/08/04]
```

nicefrac uses TeX boxes, so `\@ensuredmath` must be restored temporarily:

```
\LetLtxMacro\LWR@origUnitsNiceFrac\@UnitsNiceFrac

\DeclareRobustCommand*{\@UnitsNiceFrac}[3][]{%
\begingroup%
\LetLtxMacro\@ensuredmath\LWR@origensuredmath%
\LWR@origUnitsNiceFrac[#1]{#2}{#3}%
\endgroup%
}
```

For Mathjax:

```
\CustomizeMathJax{\newcommand{\nicefrac}[3][]{#2/#3}}
```

File 203 `lwarp-nonfloat.sty`

§ 295 Package nonfloat

(*Emulates or patches code by* Kai Rascher.)

Pkg `nonfloat` **nonfloat** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{nonfloat}

\LetLtxMacro\topcaption\caption
\newcommand{\figcaption}{\def\@captype{figure}\caption}
\newcommand{\tabcaption}{\def\@captype{table}\topcaption}
\newenvironment{narrow}[2]{}{}
```

File 204 `lwarp-nonumonpart.sty`

§ 296 Package nonumonpart

Pkg `nonumonpart` **nonumonpart** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{nonumonpart}
```

File 205 `lwarp-nopageno.sty`

§ 297 Package nopageno

Pkg `nopageno` **nopageno** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{nopageno}
```

File 206 `lwarp-nowidow.sty`

§ 298 Package nowidow

(*Emulates or patches code by* Raphaël Pinson.)

Pkg nowidow **nowidow** is not used during HTML conversion.

Discard all options for **lwarp-nowidow**:

for HTML output:
```
\LWR@ProvidesPackageDrop{nowidow}
```

\nowidow [⟨*lines*⟩]
\setnowidow [⟨*lines*⟩]

```
\newcommand*{\nowidow}[1][]{}
\newcommand*{\setnowidow}[1][]{}
```

\noclub [⟨*lines*⟩]
\setnoclub [⟨*lines*⟩]

```
\newcommand*{\noclub}[1][]{}
\newcommand*{\setnoclub}[1][]{}
```

File 207 `lwarp-ntheorem.sty`

§ 299 Package ntheorem

(*Emulates or patches code by* WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg ntheorem **ntheorem** is patched for use by **lwarp**.

Table 13: Ntheorem package — CSS styling of theorems and proofs

Theorem: `<div>` of class `theorembody<theoremstyle>`

Theorem Header: `<span>` of class `theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

§ 299.1 Limitations

⚠ Font control This conversion is not total. Font control is via css, and the custom LaTeX font settings are ignored.

⚠ Equation numbering **ntheorem** has a bug with equation numbering in $\mathcal{AMS}$ environments when the option `thref` is used. **lwarp** does not share this bug, so equations with `\split`, etc, are numbered correctly with **lwarp**'s HTML output, but not with the print output. It is recommended to use **cleveref** instead of **ntheorem**'s `thref` option.

§ 299.2 Options

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ Disabled options

The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because **cleveref** functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because **lwarp** emulated **hyperref**.

for HTML output:

Some disabled options:

```
\DeclareOption{thref}{}


\newbool{LWR@ntheoremmarks}
\boolfalse{LWR@ntheoremmarks}

\DeclareOption{thmmarks}{
\booltrue{LWR@ntheoremmarks}
\newif\ifsetendmark\setendmarktrue
}


\newbool{LWR@ntheoremamsthm}
\boolfalse{LWR@ntheoremamsthm}

\DeclareOption{amsthm}{\booltrue{LWR@ntheoremamsthm}}


\DeclareOption{amsmath}{}
\DeclareOption{hyperref}{}

\LWR@ProvidesPackagePass{ntheorem}
```

§ 299.3 Remembering the theorem style

Storage for the style being used for new theorems.

```
\newcommand{\LWR@newtheoremstyle}{plain}

\AtBeginDocument{
\@ifpackageloaded{cleveref}{
\gdef\@thm#1#2#3{%
   \if@thmmarks
     \stepcounter{end\InTheoType ctr}%
   \fi
```

```
   \renewcommand{\InTheoType}{#1}%
   \if@thmmarks
     \stepcounter{curr#1ctr}%
     \setcounter{end#1ctr}{0}%
   \fi
   \refstepcounter[#1]{#2}%  <<< cleveref modification
   \theorem@prework
    \LWR@forcenewpage% lwarp
    \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
   \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
   \ifuse@newframeskips  % cf. latex.ltx for topsepadd: \@trivlist
     \ifthm@inframe
       \thm@topsep\theoreminframepreskipamount
       \thm@topsepadd\theoreminframepostskipamount
      \else
       \thm@topsep\theorempreskipamount
       \thm@topsepadd\theorempostskipamount
      \fi
    \else% oldframeskips
      \thm@topsep\theorempreskipamount
      \thm@topsepadd \theorempostskipamount
      \ifvmode\advance\thm@topsepadd\partopsep\fi
   \fi
   \@topsep\thm@topsep
   \@topsepadd\thm@topsepadd
   \advance\linewidth -\theorem@indent
   \advance\linewidth -\theorem@rightindent
   \advance\@totalleftmargin \theorem@indent
   \parshape \@ne \@totalleftmargin \linewidth
   \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
}
}{% not @ifpackageloaded{cleveref}
\gdef\@thm#1#2#3{%
   \if@thmmarks
     \stepcounter{end\InTheoType ctr}%
   \fi
   \renewcommand{\InTheoType}{#1}%
   \if@thmmarks
     \stepcounter{curr#1ctr}%
     \setcounter{end#1ctr}{0}%
   \fi
   \refstepcounter{#2}%
   \theorem@prework
    \LWR@forcenewpage% lwarp
    \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
   \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
   \ifuse@newframeskips  % cf. latex.ltx for topsepadd: \@trivlist
     \ifthm@inframe
       \thm@topsep\theoreminframepreskipamount
       \thm@topsepadd\theoreminframepostskipamount
```

```
      \else
       \thm@topsep\theorempreskipamount
       \thm@topsepadd\theorempostskipamount
      \fi
    \else% oldframeskips
      \thm@topsep\theorempreskipamount
      \thm@topsepadd \theorempostskipamount
      \ifvmode\advance\thm@topsepadd\partopsep\fi
  \fi
  \@topsep\thm@topsep
  \@topsepadd\thm@topsepadd
  \advance\linewidth -\theorem@indent
  \advance\linewidth -\theorem@rightindent
  \advance\@totalleftmargin \theorem@indent
  \parshape \@ne \@totalleftmargin \linewidth
  \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
}
}
}% AtBeginDocument
```

Patched to remember the style being used for new theorems:

```
\gdef\theoremstyle#1{%
  \@ifundefined{th@#1}{\@warning
          {Unknown theoremstyle `#1'. Using `plain'}%
          \theorem@style{plain}
            \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
            }%
     {
       \theorem@style{#1}
       \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
       }
}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

\gdef\@xnthm#1#2[#3]{%
 \ifthm@tempif
    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
    \expandafter\@ifundefined{c@#1}%
       {\@definecounter{#1}}{}%
    \@newctr{#1}[#3]%
    \expandafter\xdef\csname the#1\endcsname{%
      \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
         {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
    \expandafter\gdef\csname mkheader@#1\endcsname
      {\csname setparms@#1\endcsname
```

```
        \@thm{#1}{#1}{#2}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
  \fi
}

\gdef\@ynthm#1#2{%
  \ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
     \expandafter\@ifundefined{c@#1}%
        {\@definecounter{#1}}{}%
     \expandafter\xdef\csname the#1\endcsname
        {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
        \@thm{#1}{#1}{#2}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
  \fi
}

\gdef\@othm#1[#2]#3{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}%
   {\ifthm@tempif
     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
     \global\@namedef{the#1}{\@nameuse{the#2}}%
     \expandafter\protected@xdef\csname num@addtheoremline#1\endcsname{%
              \noexpand\@num@addtheoremline{#1}{#3}}%
     \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
              \noexpand\@nonum@addtheoremline{#1}{#3}}%
    \theoremkeyword{#3}%
    \expandafter\protected@xdef\csname #1Keyword\endcsname
             {\the\theoremkeyword}%
     \expandafter\gdef\csname mkheader@#1\endcsname
       {\csname setparms@#1\endcsname
                \@thm{#1}{#2}{#3}
        }%
     \global\@namedef{end#1}{\@endtheorem}
     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
  \fi}
}
```

§ 299.4 HTML cross-referencing

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```
\newcommand{\LWR@inctheorem}{%
\addtocounter{LWR@thisautoid}{1}%
\LWR@stoppars%
\LWR@htmltag{a id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
\LWR@startpars%
}
```

§ 299.5 `\newtheoremstyle`

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader<style>`.

```
\gdef\newtheoremstyle#1#2#3{%
  \expandafter\@ifundefined{th@#1}%
   {\expandafter\gdef\csname th@#1\endcsname{%
    \def\@begintheorem####1####2{%
    \LWR@inctheorem% lwarp
    #2}%
    \def\@opargbegintheorem####1####2####3{%
    \LWR@inctheorem% lwarp
    #3}%
}%
}%
{\PackageError{\basename}{Theorem style #1 already defined}\@eha}
}
```

§ 299.6 Standard styles

```
\renewtheoremstyle{plain}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}

\renewtheoremstyle{break}%
  {\item[
    \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
    ]}%
```

```
  {\item[
    \InlineClass{theoremheaderbreak}%
        {##1\ ##2\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{change}%
  {\item[
    \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
  {\item[
    \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}

\renewtheoremstyle{changebreak}%
  {\item[
        \InlineClass{theoremheaderchangebreak}%
            {##2\ ##1\theorem@separator}\newline
    ]}%
  {\item[
        \InlineClass{theoremheaderchangebreak}%
            {##2\ ##1\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{margin}%
  {\item[
        \InlineClass{theoremheadermargin}{##2 \qquad ##1\theorem@separator}
    ]}%
  {\item[
        \InlineClass{theoremheadermargin}{##2 \qquad ##1\ (##3)\theorem@separator}
    ]}

\renewtheoremstyle{marginbreak}%
  {\item[
    \InlineClass{theoremheadermarginbreak}%
        {##2 \qquad ##1\theorem@separator}\newline
    ]}%
  {\item[
    \InlineClass{theoremheadermarginbreak}%
        {##2 \qquad ##1\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{nonumberplain}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
  {\item[
    \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}

\renewtheoremstyle{nonumberbreak}%
  {\item[
    \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
    ]}%
  {\item[
```

```
    \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
    ]}

\renewtheoremstyle{empty}%
  {\item[]}%
  {\item[
    \InlineClass{theoremheaderplain}{##3}]}

\renewtheoremstyle{emptybreak}%
  {\item[]}%
  {\item[
    \InlineClass{theoremheaderplain}{##3}] \ \newline}
```

§ 299.7 Additional objects

The following manually adjust the CSS for the standard configuration objects which are not a purely plain style:

```
\ifbool{LWR@ntheoremamsthm}{}{%
```

Upright text via CSS:

```
    \newtheoremstyle{plainupright}%
    {\item[
        \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
    {\item[
        \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
```

Upright text and small caps header via CSS:

```
    \newtheoremstyle{nonumberplainuprightsc}%
    {\item[
        \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
    {\item[
        \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}
}% not amsthm
```

§ 299.8 Renewed standard configuration

The following standard configuration is renewed using the new CSS:

```
\ifbool{LWR@ntheoremamsthm}{}{%


\ifx\thm@usestd\@undefined
\else
    \theoremnumbering{arabic}
    \theoremstyle{plain}
    \RequirePackage{latexsym}
```

```
    \theoremsymbol{\Box}
    \theorembodyfont{\itshape}
    \theoremheaderfont{\normalfont\bfseries}
    \theoremseparator{}
    \renewtheorem{Theorem}{Theorem}
    \renewtheorem{theorem}{Theorem}
    \renewtheorem{Satz}{Satz}
    \renewtheorem{satz}{Satz}
    \renewtheorem{Proposition}{Proposition}
    \renewtheorem{proposition}{Proposition}
    \renewtheorem{Lemma}{Lemma}
    \renewtheorem{lemma}{Lemma}
    \renewtheorem{Korollar}{Korollar}
    \renewtheorem{korollar}{Korollar}
    \renewtheorem{Corollary}{Corollary}
    \renewtheorem{corollary}{Corollary}

    \theoremstyle{plainupright}
    \theorembodyfont{\upshape}
    \theoremsymbol{\HTMLunicode{25A1}}% UTF-8 white box
    \renewtheorem{Example}{Example}
    \renewtheorem{example}{Example}
    \renewtheorem{Beispiel}{Beispiel}
    \renewtheorem{beispiel}{Beispiel}
    \renewtheorem{Bemerkung}{Bemerkung}
    \renewtheorem{bemerkung}{Bemerkung}
    \renewtheorem{Anmerkung}{Anmerkung}
    \renewtheorem{anmerkung}{Anmerkung}
    \renewtheorem{Remark}{Remark}
    \renewtheorem{remark}{Remark}
    \renewtheorem{Definition}{Definition}
    \renewtheorem{definition}{Definition}

    \theoremstyle{nonumberplainuprightsc}
    \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
    \renewtheorem{Proof}{Proof}
    \renewtheorem{proof}{Proof}
    \renewtheorem{Beweis}{Beweis}
    \renewtheorem{beweis}{Beweis}
    \qedsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof

    \theoremsymbol{}
\fi
}% not amsthm
```

§ 299.9 amsthm option

Only if the amsthm option was given:

```
\ifbool{LWR@ntheoremamsthm}{

\gdef\th@plain{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% lwarp
      \item[
  \InlineClass{theoremheaderplain}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% lwarp
     \item[
  \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberplain{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% lwarp
      \item[
  \InlineClass{theoremheaderplain}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% lwarp
     \item[
  \InlineClass{theoremheaderplain}{##1\ (##3).}
        ]}}

\gdef\th@definition{%
  \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% lwarp
      \item[
  \InlineClass{theoremheaderdefinition}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% lwarp
     \item[
  \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberdefinition{%
  \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% lwarp
      \item[
```

```
  \InlineClass{theoremheaderdefinition}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% lwarp
     \item[
  \InlineClass{theoremheaderdefinition}{##1\ (##3).}
        ]}}

\gdef\th@remark{%
  \def\theorem@headerfont{\itshape}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% lwarp
      \item[
  \InlineClass{theoremheaderremark}{##1\ ##2.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% lwarp
     \item[
  \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
        ]}}

\gdef\th@nonumberremark{%
  \def\theorem@headerfont{\itshape}\normalfont%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% lwarp
      \item[
  \InlineClass{theoremheaderremark}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% lwarp
     \item[
  \InlineClass{theoremheaderremark}{##1\ (##3).}
        ]}}

\gdef\th@proof{%
  \def\theorem@headerfont{\normalfont\bfseries}\itshape%
  \def\@begintheorem##1##2{%
        \LWR@inctheorem% lwarp
      \item[
  \InlineClass{theoremheaderproof}{##1.}
        ]}%
  \def\@opargbegintheorem##1##2##3{%
        \LWR@inctheorem% lwarp
     \item[
  \InlineClass{theoremheaderproof}{##1\ (##3).}
        ]}}



\newcounter{proof}%
```

```
\if@thmmarks
    \newcounter{currproofctr}%
    \newcounter{endproofctr}%
\fi

\gdef\proofSymbol{\openbox}

\newcommand{\proofname}{Proof}

\newenvironment{proof}[1][\proofname]{
    \th@proof
    \def\theorem@headerfont{\itshape}%
    \normalfont
    \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
    \@thm{proof}{proof}{#1}
}%
{\@endtheorem}

}{}% amsthm option
```

§ 299.10 Ending a theorem

Patched for CSS:

```
\let\LWR@origendtheorem\@endtheorem
\renewcommand{\@endtheorem}{%
\ifbool{LWR@ntheoremmarks}{%
    \ifsetendmark%
    \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
    \setendmarkfalse%
    \fi%
}{}%
\LWR@origendtheorem% also does \@endtrivlist
\ifbool{LWR@ntheoremmarks}{\global\setendmarktrue}{}%
\endBlockClass%
}
```

§ 299.11 `\NoEndMark`

```
\gdef\NoEndMark{\global\setendmarkfalse}
```

§ 299.12 List-of

Redefined to reuse the float mechanism to add list-of-theorem links:

`\thm@thmline` {⟨*1: printed type*⟩} {⟨*2: #*⟩} {⟨*3: optional*⟩} {⟨*4: page*⟩}

```
\renewcommand{\thm@@thmline@noname}[4]{%
\hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
```

```
}

\renewcommand{\thm@@thmline@name}[4]{%
\hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
}
```

This was redefined by **ntheorem** when loaded, so it is now redefined for **lwarp**:

```
\def\thm@@thmline{\thm@@thmline@name}
```

Patch for CSS:

```
\def\listtheorems#1{
\LWR@htmlelementclass{nav}{lothm}%
\begingroup
\c@tocdepth=-2%
\def\thm@list{#1}\thm@processlist
\endgroup
\LWR@htmlelementclassend{nav}{lothm}%
}
```

§ 299.13 Symbols

Proof QED symbol:

```
\newcommand{\qed}{\qquad\the\qedsymbol}

\AtBeginDocument{
\@ifundefined{LWR@orig@openbox}{
\LetLtxMacro\LWR@orig@openbox\openbox
\LetLtxMacro\LWR@orig@blacksquare\blacksquare
\LetLtxMacro\LWR@orig@Box\Box

\def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
\def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
\def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\openbox\LWR@orig@openbox%
\LetLtxMacro\blacksquare\LWR@orig@blacksquare%
\LetLtxMacro\Box\LWR@orig@Box%
}% appto
}{}% @ifundefined
}% AtBeginDocument
```

§ 299.14 Cross-referencing

`\thref {`⟨*label*⟩`}`

```
481 \newcommand*{\thref}[1]{\cref{#1}}
```

File 208 `lwarp-overpic.sty`

§ 300 Package overpic

(*Emulates or patches code by* Rolf Niepraschk.)

Pkg `overpic` **overpic** is patched for use by **lwarp**.

⚠ scaling The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

See section 80.2 for the print-mode version of `\overpicfontsize` and `\overpicfontskip`.

for HTML output:

```
1 \LWR@ProvidesPackagePass{overpic}

2 \newcommand*{\overpicfontsize}{12}
3 \newcommand*{\overpicfontskip}{14}
4
5 \BeforeBeginEnvironment{overpic}{%
6     \begin{lateximage}%
7     \fontsize{\overpicfontsize}{\overpicfontskip}%
8     \selectfont%
9 }
10
11 \AfterEndEnvironment{overpic}{\end{lateximage}}
12
13 \BeforeBeginEnvironment{Overpic}{%
14     \begin{lateximage}%
15     \fontsize{\overpicfontsize}{\overpicfontskip}%
16     \selectfont%
17 }
18
19 \AfterEndEnvironment{Overpic}{\end{lateximage}}
```

File 209 `lwarp-pagegrid.sty`

§ 301 Package pagegrid

Pkg `pagegrid` **pagegrid** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{pagegrid}
\newcommand*{\pagegridsetup}[1]{}
```

File 210 `lwarp-pagenote.sty`

§ 302 Package pagenote

Pkg `pagenote` **pagenote** works as-is, but the `page` option is disabled.

for HTML output:

```
\DeclareOption{page}{}
\LWR@ProvidesPackagePass{pagenote}
```

File 211 `lwarp-pagesel.sty`

§ 303 Package pagesel

Pkg `pagesel` **pagesel** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{pagesel}
```

File 212 `lwarp-paralist.sty`

§ 304 Package paralist

(*Emulates or patches code by* Bernd Schandl.)

Pkg `paralist` **paralist** is supported with minor changes.

for HTML output:

```
\LWR@ProvidesPackagePass{paralist}
```

The compact environments are identical to the regular ones:

```
\LetLtxMacro\compactitem\itemize
\LetLtxMacro\compactenum\enumerate
\LetLtxMacro\compactdesc\description
\LetLtxMacro\endcompactitem\enditemize
\LetLtxMacro\endcompactenum\endenumerate
\LetLtxMacro\endcompactdesc\enddescription
```

For the inline environments, revert `\item` to its original print-mode version:

```
\AtBeginEnvironment{inparaitem}{\LetLtxMacro\item\LWR@origitem}
\AtBeginEnvironment{inparaenum}{\LetLtxMacro\item\LWR@origitem}
\AtBeginEnvironment{inparadesc}{\LetLtxMacro\item\LWR@origitem}
```

Manual formatting of the description labels:

```
\def\paradescriptionlabel#1{{\normalfont\textbf{#1}}}
```

File 213 `lwarp-parnotes.sty`

§ 305 Package

parnotes

(*Emulates or patches code by* Chelsea Hughes.)

Pkg `parnotes` **parnotes** is supported with some patches.

for HTML output:

```
\LWR@ProvidesPackagePass{parnotes}

\long\def\PN@parnote@real#1#2{%
    \parnotemark{#1}%
    % Unless this is the first parnote in \PN@text, add a separator first
    \unless\ifx\PN@text\@empty\g@addto@macro\PN@text{\parnoteintercmd}\fi
    % Redefine \@currentlabel to the parnote label, so \label works
    \g@addto@macro\PN@text{%
%         \phantomsection%
        \def\@currentlabel{#1}%
        \def\cref@currentlabel{%        lwarp
            [parnotemark][\arabic{parnotemark}][]\theparnotemark%
        }%
    }%
    \g@addto@macro\PN@text{%
        \LWR@textcurrentfont{%          lwarp
            \parnotemark{#1}\nolinebreak\thinspace#2%
        }%
    }%
```

```
}

\def\PN@parnotes@real{%
    % We call \par later, so this avoids recursion with \PN@parnotes@auto
    \PN@inparnotestrue
    \unless\ifvmode\par\fi
    % Avoid page breaks between a paragraph and its parnotes
    \nopagebreak\addvspace{\parnotevskip}%
    \LWR@forcenewpage%                  lwarp
    \begin{BlockClass}{footnotes}%      lwarp
    {\parnotefmt{\PN@text}\par}%
    \end{BlockClass}%                   lwarp
    \global\def\PN@text{}%
    \addvspace{\parnotevskip}%
    %
    % These can be enabled or disabled by package options
    %
    \PN@disable@indent
    \PN@reset@optional
    \PN@inparnotesfalse
}

\AtBeginDocument{
\crefname{parnotemark}{paragraph note}{paragraph notes}
\Crefname{parnotemark}{Paragraph note}{Paragraph notes}
}
```

File 214 `lwarp-parskip.sty`

§ 306 Package parskip

Pkg `parskip` **parskip** is ignored.

for HTML output: Discard all options for **lwarp-parskip**.

```
\LWR@ProvidesPackageDrop{parskip}
```

File 215 `lwarp-pbox.sty`

§ 307 Package pbox

(*Emulates or patches code by* Simon Law.)

Pkg `pbox` **pbox** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{pbox}

\NewDocumentCommand{\pbox}{O{t} o O{t} m +m}{%
\booltrue{LWR@minipagefullwidth}%
\parbox[#1][#2][#3]{#4}{#5}%
}

\newcommand{\settominwidth}[3][\columnwidth]{%
\settowidth{#2}{#3}%
}

\newcommand{\widthofpbox}[1]{%
\widthof{#1}%
}
```

File 216 `lwarp-pdflscape.sty`

§ 308 Package pdflscape

Pkg pdflscape **pdflscape** is ignored.

for HTML output: Discard all options for **lwarp-pdflscape**:

```
\LWR@ProvidesPackageDrop{pdflscape}
```

File 217 `lwarp-pdfpages.sty`

§ 309 Package pdfpages

(*Emulates or patches code by* Andreas Matthias.)

Pkg pdfpages **pdfpages** is patched for use by **lwarp**.

Option `link` and `linkname` work:

```
\hyperlink{<filename>.pdf.<pagenubmer}}{some text}
\hyperlink{<linkname>.<pagenubmer}}{some text}
```

Options which make no sense in HTML are disabled.

for HTML output:

```
\LWR@ProvidesPackagePass{pdfpages}
```

Disable option which have no meaning for HTML output:

```
\define@key{pdfpages}{fitpaper}[false]{}
\define@key{pdfpages}{landscape}[false]{}
\define@key{pdfpages}{openright}[false]{}
\define@key{pdfpages}{signature}{}
\define@key{pdfpages}{signature*}{}
\define@key{pdfpages}{booklet}[false]{}
\define@key{pdfpages}{rotateoversize}[false]{}
\define@key{pdfpages}{doublepages}[false]{}
\define@key{pdfpages}{doublepagestwist}[false]{}
\define@key{pdfpages}{doublepagestwistodd}[false]{}
\define@key{pdfpages}{doublepagestwist*}[false]{}
\define@key{pdfpages}{doublepagestwistodd*}[false]{}
\define@key{pdfpages}{duplicatepages}[2]{}
\define@key{pdfpages}{thread}[false]{}
\define@key{pdfpages}{threadname}{}
\define@key{pdfpages}{linkfit}{}
\define@key{pdfpages}{linktodoc}[false]{}
\define@key{pdfpages}{linktodocfit}{}
\define@key{pdfpages}{linkfilename}{}
\define@key{pdfpages}{survey}[false]{}
\define@key{pdfpages}{survey-nolink}[false]{}
\define@key{pdfpages}{newwindow}[false]{}
```

Use print mode while measuring the page numbers:

```
\xpretocmd{\AM@getpagecount}{\LWR@restoreorigformatting}{}{}
```

Emulate a bit of **eso-pic**:

```
\newif\ifESO@texcoord

\newcommand{\ESO@HookIIBG}{}

\renewcommand{\AM@AddToShipoutPicture}{\g@addto@macro\ESO@HookIIBG}

\renewcommand{\ClearShipoutPicture}{}
```

`\LWR@esopic@newpage` At each `\newpage`.

```
\newcommand*{\LWR@esopic@newpage}{%
```

Is there something to draw?

```
\ifdefvoid{\ESO@HookIIBG}%
{}%
{%
```

If the `link` option was specified, add a hyper taraget:

```
    \ifAM@link%
        \hypertarget{\AM@linkname.\AM@page}{}%
    \fi%
```

Draw inside a picture environment of the size of a virtual page:

```
    \begingroup%
    \setlength{\unitlength}{1in}%
    \begin{picture}(8,10.5)%
    \ESO@HookIIBG%
    \end{picture}%
    \endgroup%
    \global\let\ESO@HookIIBG\@empty%
}
}
```

`\AM@output` Patched to use `\LWR@esopic@newpage`.

```
\xpatchcmd{\AM@output}
    {\newpage}
    {\LWR@esopic@newpage}
    {}
    {\LWR@patcherror{pdfpages}{AM@output-1}}

\xpatchcmd{\AM@output}
    {\newpage}
    {\LWR@esopic@newpage}
    {}
    {\LWR@patcherror{pdfpages}{AM@output-2}}

\xpatchcmd{\AM@output}
    {\newpage}
    {\LWR@esopic@newpage}
    {}
    {\LWR@patcherror{pdfpages}{AM@output-3}}
```

`\includepdf` Patched to set a reasonable paper size.

```
\xpretocmd{\includepdf}{%
    \begingroup%
    \setlength{\paperwidth}{8in}%
    \setlength{\paperheight}{10.5in}%
}{}{}

\xapptocmd{\includepdf}{%
    \endgroup%
}{}{}
```

\includepdfmerge Patched to set a reasonable paper size.

```
\xpretocmd{\includepdfmerge}{%
    \begingroup%
    \setlength{\paperwidth}{8in}%
    \setlength{\paperheight}{10.5in}%
}{}{}

\xapptocmd{\includepdfmerge}{%
    \endgroup%
}{}{}
```

\AM@hyper@begin@i Hyper links are created by \LWR@esopic@newpage, so don't create them here:

```
\renewcommand{\AM@hyper@begin@i}{}
```

File 218 `lwarp-pdfrender.sty`

§ 310 Package pdfrender

Pkg pdfrender **pdfrender** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{pdfrender}

\newcommand*{\pdfrender}[1]{}
\newcommand{\textpdfrender}[2]{#2}
```

File 219 `lwarp-pdfsync.sty`

§ 311 Package pdfsync

(*Emulates or patches code by* J. Laurens.)

Pkg pdfsync Emulated.

for HTML output: Discard all options for **lwarp-pdfsync**:

```
\LWR@ProvidesPackageDrop{pdfsync}

\newcommand*{\pdfsync}{}
\newcommand*{\pdfsyncstart}{}
\newcommand*{\pdfsyncstop}{}
```

File 220 `lwarp-pdftricks.sty`

§ 312 Package

pdftricks

(*Emulates or patches code by* C. V. Radhakrishnan, C. V. Rajagopal, Antoine Chambert-Loir.)

Pkg `pdftricks` **pdftricks** is patched for use by **lwarp**.

⚠ convert image files The **pdftricks** image files `<jobname>-fig*.pdf` must be converted to `.svg`, or else a missing file error will occur. The image files must also be converted again whenever they change. To convert the images:

Enter ⇒ `lwarpmk pdftosvg <jobname>-fig*.pdf`

for HTML output:

```
\LWR@ProvidesPackagePass{pdftricks}
```

Reuse the print-mode images:

```
\def\PDFTfigname{\BaseJobname-fig\thepsfig}
```

If the `.pdf` images have not yet been converted to `.svg` then an error about a missing file will occur. Warn the user to convert the images.

```
\PackageWarning{lwarp-pdftricks}{
When the pdftricks images change,
remember to convert PDF images to SVG using 'lwarpmk pdftosvg *-fig.pdf',
}

\AfterEndDocument{\typeout{***}}
\AfterEndDocument{\typeout{*** Note: If pdftricks images are not found, new, or updated,}}
\AfterEndDocument{\typeout{*** \space use 'lwarpmk pdftosvg \BaseJobname-fig*.pdf'}}
\AfterEndDocument{\typeout{***}}
```

File 221 `lwarp-pdfx.sty`

§ 313 Package

pdfx

Pkg `pdfx` **pdfx** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{pdfx}
```

File 222 `lwarp-pfnote.sty`

§ 314 Package pfnote

Pkg `pfnote` **pfnote** is emulated.

⚠ **pfnote** numbers While emulating **pfnote**, **lwarp** is not able to reset HTML footnote numbers per page number to match the printed version, as HTML has no concept of page numbers. **lwarp** therefore uses continuous footnote numbering even for **pfnote**.

for HTML output:

```
\LWR@ProvidesPackageDrop{pfnote}
```

File 223 `lwarp-phfqit.sty`

§ 315 Package phfqit

(*Emulates or patches code by* Philippe Faist.)

Pkg `phfqit` **phfqit** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{phfqit}

\LetLtxMacro\LWR@origbitstring\bitstring

\renewcommand\bitstring[1]{%
\InlineClass[%
    text-decoration: overline underline ;
]{bitstring}{#1}%
% \phfqit@bitstring{#1}%
}

\appto\LWR@restoreorigformatting{%
\LetLtxMacro\bitstring\LWR@origbitstring%
}
```

File 224 `lwarp-placeins.sty`

§ 316 Package placeins

(*Emulates or patches code by* Donald Arseneau.)

Pkg placeins **placeins** is not used during HTML conversion.

Discard all options for **lwarp-placeins**:

for HTML output:

```
\LWR@ProvidesPackageDrop{placeins}
\newcommand*{\FloatBarrier}{}
```

File 225 `lwarp-prelim2e.sty`

§ 317 Package prelim2e

(*Emulates or patches code by* Martin Schröder.)

Pkg prelim2e Emulated.

for HTML output: Discard all options for **lwarp-prelim2e**:

```
\LWR@ProvidesPackageDrop{prelim2e}
\newcommand{\PrelimText}{}
\newcommand{\PrelimTextStyle}{}
\newcommand{\PrelimWords}{}
```

File 226 `lwarp-prettyref.sty`

§ 318 Package prettyref

(*Emulates or patches code by* Kevin S. Ruland.)

Pkg prettyref **prettyref** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{prettyref}
\newrefformat{fig}{Figure \ref{#1}}
\newrefformat{tab}{Table \ref{#1}}
```

File 227 `lwarp-preview.sty`

§ 319 Package preview

Pkg preview **preview** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{preview}

\newenvironment{preview}{}{}
\newenvironment{nopreview}{}{}
\NewDocumentCommand{\PreviewMacro}{s o o +m}{}
\NewDocumentCommand{\PreviewEnvironment}{s o o +m}{}
\newcommand{\PreviewSnarfEnvironment}[2][]{}
\NewDocumentCommand{\PreviewOpen}{s o}{}
\NewDocumentCommand{\PreviewClose}{s o}{}
\let\ifPreview\iffalse% \fi for syntax highlighting
```

File 228 `lwarp-pst-eps.sty`

§ 320 Package pst-eps

(*Emulates or patches code by* Herbert Voss.)

Pkg `pst-eps` **pst-eps** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{pst-eps}

\renewenvironment{TeXtoEPS}{}{}
\renewcommand{\PSTtoEPS}[3][]{}
```

File 229 `lwarp-pstricks.sty`

§ 321 Package pstricks

(*Emulates or patches code by* Timothy Van Zandt.)

Pkg `pstricks` **pstricks** is patched for use by **lwarp**.

⚠ use `pspicture` All **pstricks** content should be contained inside a `pspicture` environment.

for HTML output:

```
\LWR@ProvidesPackagePass{pstricks}

\BeforeBeginEnvironment{pspicture}{\begin{lateximage}[(pspicture)]}
\AfterEndEnvironment{pspicture}{\end{lateximage}}
```

File 230 `lwarp-quotchap.sty`

§ 322 Package

quotchap

(*Emulates or patches code by* Karsten Tinnefeld, Jan Klever.)

Pkg quotchap **quotchap** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{quotchap}

\newcommand{\@quotchap}{}
\newlength{\LWR@quotchapwidth}

\let\@printcites\relax

\newcommand*{\@iprintcites}{%
```

Place the quotes inside a `<div>` of class `quotchap`, of the maximum selected width:

```
\begin{BlockClass}[max-width: \LWR@printlength{\LWR@quotchapwidth}]{quotchap}
%\begin{minipage}{\LWR@quotchapwidth}
\@quotchap
%\end{minipage}
\end{BlockClass}
```

Deactivate the quote printing:

```
\global\let\@printcites\relax
}

\NewEnviron{savequote}[1][\linewidth]{%
```

Remember the width, adjusted for HTML, and make the length assignment global, per:
https://tex.stackexchange.com/questions/300823/why-is-setlength-ineffective-inside-a-tabular-environment

```
\setlength{\LWR@quotchapwidth}{#1*2}%
\global\LWR@quotchapwidth=\LWR@quotchapwidth%
```

Remember the body, and activate the quote printing:

```
\global\let\@quotchap\BODY
\global\let\@printcites\@iprintcites%
```

```
}
```

The quotation author is placed inside a of class qauthor:

```
\newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

\qsetcnfont is ignored:

```
\newcommand{\qsetcnfont}[1]{}
```

File 231 lwarp-quoting.sty

§ 323 Package quoting

(*Emulates or patches code by* Thomas Titz.)

Pkg quoting **quoting** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{quoting}

\xpatchcmd{\quoting}{\quo@begintext}
    {\begin{LWR@blocktextcurrentfont}\quo@begintext}
    {}
    {\LWR@patcherror{quoting}{quoting}}

\xpatchcmd{\endquoting}{\quo@endtext}
    {\quo@endtext\end{LWR@blocktextcurrentfont}}
    {}
    {\LWR@patcherror{imakeidx}{endquoting}}
```

File 232 lwarp-ragged2e.sty

§ 324 Package ragged2e

(*Emulates or patches code by* Martin Schröder.)

Pkg ragged2e **ragged2e** is not used during HTML conversion.

Discard all options for **lwarp-ragged2e**:

for HTML output:

```
\LWR@ProvidesPackageDrop{ragged2e}

\LetLtxMacro\Centering\centering
```

```
\LetLtxMacro\RaggedLeft\raggedleft
\LetLtxMacro\RaggedRight\raggedright
\newcommand*{\justifying}{}
\newlength{\CenteringLeftskip}
\newlength{\RaggedLeftLeftskip}
\newlength{\RaggedRightLeftskip}
\newlength{\CenteringRightskip}
\newlength{\RaggedLeftRightskip}
\newlength{\RaggedRightRightskip}
\newlength{\CenteringParfillskip}
\newlength{\RaggedLeftParfillskip}
\newlength{\RaggedRightParfillskip}
\newlength{\JustifyingParfillskip}
\newlength{\CenteringParindent}
\newlength{\RaggedLeftParindent}
\newlength{\RaggedRightParindent}
\newlength{\JustifyingParindent}
\newenvironment*{Center}{\center}{\endcenter}
\newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
\newenvironment*{FlushRight}{\flushright}{\endflushright}
\newenvironment*{justify}{\justifying}{\endjustifying}
```

File 233 `lwarp-realscripts.sty`

§ 325 Package realscripts

(*Emulates or patches code by* Will Robertson.)

Pkg `realscripts`

realscripts is emulated. See `lwarp.css` for the `<span>` of class `supsubscript`.

for HTML output:

```
\LWR@ProvidesPackageDrop{realscripts}

\let\realsuperscript\textsuperscript
\let\realsubscript\textsubscript

\let\fakesuperscript\textsuperscript
\let\fakesubscript\textsubscript

\newlength{\subsupersep}

\newcommand*{\LWR@realscriptsalign}{}

\newcommand*{\LWR@setrealscriptsalign}[1]{%
\renewcommand*{\LWR@realscriptsalign}{}%
\ifthenelse{\equal{#1}{c}}{%
    \renewcommand{\LWR@realscriptsalign}{\LWR@print@mbox{text-align:center} ; }%
}{}%
```

```
\ifthenelse{\equal{#1}{r}}{%
    \renewcommand{\LWR@realscriptsalign}{\LWR@print@mbox{text-align:right} ; }%
}{}%
}

\DeclareDocumentCommand \textsubsuperscript {s O{l} mm} {%
\LWR@setrealscriptsalign{#2}%
\InlineClass[\LWR@realscriptsalign]{supsubscript}{%
\textsuperscript{#4}\textsubscript{#3}%
}%
}

\DeclareDocumentCommand \textsupersubscript {s O{l} mm} {%
\LWR@setrealscriptsalign{#2}%
\InlineClass[\LWR@realscriptsalign]{supsubscript}{%
\textsubscript{#4}\textsuperscript{#3}%
}%
}
```

File 234 `lwarp-register.sty`

§ 326 Package register

(*Emulates or patches code by* Matthew Lovell.)

Pkg `register` **register** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{register}

\xpatchcmd{\register}
    {\centering}
    {\begin{center}\begin{lateximage}[-register-~\packagediagramname]}
    {}
    {\LWR@patcherror{register}{register}}

\xpatchcmd{\endregister}
    {\leftskip}
    {%
        \end{lateximage}\end{center}%
        \leftskip%
    }%
    {}
    {\LWR@patcherror{register}{endregister}}

\setlength{\regWidth}{5in}
```

File 235 `lwarp-relsize.sty`

§ 327 Package

relsize

(*Emulates or patches code by* Donald Arseneau, Bernie Cosell, Matt Swift.)

Pkg `relsize` **relsize** is patched for use by **lwarp**.

For HTML only the inline macros are supported: `\textlarger`, `\textsmaller`, and `\textscale`. Each becomes an inline span of a modified `font-size`.

`\relsize`, `\larger`, `\smaller`, and `\relscale` are ignored.

While creating SVG math for HTML, the original definitions are temporarilty restored, and so should work as expected.

⚠ not small The HTML browser's setting for minumum font size may limit how small the output will be displayed.

for HTML output:

```
\LWR@ProvidesPackagePass{relsize}

\let\LWR@origrelsize\relsize
\LetLtxMacro\LWR@origlarger\larger
\LetLtxMacro\LWR@origsmaller\smaller
\let\LWR@relscale\relscale
\LetLtxMacro\LWR@origtextlarger\textlarger
\LetLtxMacro\LWR@origtextsmaller\textsmaller
\let\LWR@textscale\textscale

\appto\LWR@restoreorigformatting{%
\let\relsize\LWR@origrelsize%
\LetLtxMacro\larger\LWR@origlarger%
\LetLtxMacro\smaller\LWR@origsmaller%
\let\relscale\LWR@relscale%
\LetLtxMacro\textlarger\LWR@origtextlarger%
\LetLtxMacro\textsmaller\LWR@origtextsmaller%
\let\textscale\LWR@textscale%
}

\newcounter{LWR@relsizetemp}

\renewcommand*{\relsize}[1]{}
\renewcommand*{\larger}[1][]{}
\renewcommand*{\smaller}[1][]{}
\renewcommand*{\relscale}[1]{}
```

```
26
27 \renewcommand*{\textlarger}[2][1]{%
28 \setcounter{LWR@relsizetemp}{100+(#1*20)}%
29 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textlarger}{#2}%
30 }
31
32 \renewcommand*{\textsmaller}[2][1]{%
33 \setcounter{LWR@relsizetemp}{100-(#1*20)}%
34 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textsmaller}{#2}%
35 }
36
37 \renewcommand*{\textscale}[2]{%
38 \setcounter{LWR@relsizetemp}{100*\real{#1}}%
39 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textscale}{#2}%
40 }
```

File 236 `lwarp-repeatindex.sty`

§ 328 Package repeatindex

Pkg `repeatindex`

repeatindex is emulated for **lwarp**.

lwarp must be used with a special style file:

```
\usepackage[makeindex,makeindexStyle={lwarp_repeatindex}]{lwarp}
```

where `lwarp_repeatindex.ist` may be copied from the following modified version of `lwarp.ist`:

```
preamble
"\\begin{theindex}
  \\providecommand*\\lettergroupDefault[1]{}
  \\providecommand*\\lettergroup[1]{%
      \\par\\textbf{#1}\\par
      \\nopagebreak
  }
"
headings_flag 1
heading_prefix "
  \\lettergroup{"
heading_suffix "}"
delim_0 "], \\hyperindexref{"
delim_1 ", \\hyperindexref{"
delim_2 ", \\hyperindexref{"
delim_n "}, \\hyperindexref{"
```

```
delim_r "} -- \\hyperindexref{"
delim_t "}"

item_0 "\n  \\item ["
```

(The modifications are the `delim_0` and `item_0` entries.)

for HTML output:

```
\LWR@ProvidesPackageDrop{repeatindex}[2001/10/13]
```

In the **lwarp** core, `\LWR@indexitem` is modified to accept the optional `\item` argument.

```
\RequirePackage{makeidx}
\def\entryprefix{\itshape}
\def\entrypostfix{\dots}
```

File 237 `lwarp-resizegather.sty`

§ 329 Package resizegather

Pkg `resizegather` **resizegather** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{resizegather}

\newcommand*{\resizegathersetup}[1]{}
```

File 238 `lwarp-romanbar.sty`

§ 330 Package romanbar

(*Emulates or patches code by* H.-Martin Münch.)

Pkg `romanbar` **romanbar** is patched for use by **lwarp**.

An inline class with an overline and underline is used.

for HTML output:

```
\LWR@ProvidesPackagePass{romanbar}

\DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
\InlineClass[%
    text-decoration: overline underline ;
]{romanbar}{#1}%
}
```

File 239 `lwarp-romanbarpagenumber.sty`

§ 331 Package romanbarpagenumber

Pkg `romanbarpagenumber` **romanbarpagenumber** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{romanbarpagenumber}
```

File 240 `lwarp-rotating.sty`

§ 332 Package rotating

(*Emulates or patches code by* Robin Fairbairns, Sebastian Rahtz, Leonor Barroca.)

Pkg `rotating` **rotating** is emulated.

All rotations are ignored in HTML output.

for HTML output:

```
\LWR@ProvidesPackageDrop{rotating}

\LetLtxMacro\sidewaystable\table
\let\endsidewaystable\endtable

\LetLtxMacro\sidewaysfigure\figure
\let\endsidewaysfigure\endfigure

\newenvironment*{sideways}{}{}
\newenvironment*{turn}[1]{}{}
\newenvironment*{rotate}[1]{}{}
\NewDocumentCommand{\turnbox}{m +m}{#2}
\let\rotcaption\caption
\let\@makerotcaption\@makecaption
```

File 241 `lwarp-rotfloat.sty`

§ 333 Package rotfloat

(*Emulates or patches code by* Axel Sommerfeldt.)

Pkg `rotfloat` **rotfloat** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{rotfloat}[2004/01/04]

\RequirePackage{float}
```

`\newfloat` {⟨*1: type*⟩} {⟨*2: placement*⟩} {⟨*3: ext*⟩} [⟨*4: within*⟩]

Emulates the `\newfloat` command from the **float** package. Sideways floats are `\let` to the same as regular floats.

"placement" is ignored.

```
\RenewDocumentCommand{\newfloat}{m m m o}{%
\IfValueTF{#4}%
{%
    \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
}%
{%
    \DeclareFloatingEnvironment[fileext=#3]{#1}%
}%
\csletcs{sideways#1}{#1}%
\csletcs{endsideways#1}{end#1}%
```

Remember the float style:

```
\csedef{LWR@floatstyle@#1}{\LWR@floatstyle}%
\csedef{LWR@floatstyle@sideways#1}{\LWR@floatstyle}%
```

newfloat package automatically creates the `\listof` command for new floats, but **float** does not, so remove `\listof` here in case it is manually created later:

```
\cslet{listof#1s}\relax%
\cslet{listof#1es}\relax%
\cslet{listofsideways#1s}\relax%
\cslet{listofsideways#1es}\relax%
}
```

File 242 `lwarp-savetrees.sty`

§ 334 Package savetrees

Pkg `savetrees` Emulated.

for HTML output: Discard all options for **lwarp-savetrees**:

```
\LWR@ProvidesPackageDrop{savetrees}
```

File 243 `lwarp-scalefnt.sty`

§ 335 Package scalefnt

(*Emulates or patches code by* D. Carlisle.)

Pkg `scalefnt` **scalefnt** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{scalefnt}

\DeclareRobustCommand\scalefont[1]{}
```

File 244 `lwarp-schemata.sty`

§ 336 Package schemata

(*Emulates or patches code by* Charles P. Schaum.)

Pkg `schemata` **schemata** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{schemata}

\LetLtxMacro\LWR@schemata@origschema\schema
\LetLtxMacro\LWR@schemata@origSchema\Schema

\renewcommand{\schema}[3][open]{%
\begin{lateximage}%
\LWR@print@normalsize
\LWR@schemata@origschema[#1]{#2}{#3}%
\end{lateximage}%
}

\renewcommand{\Schema}[5][open]{%
\begin{lateximage}%
\LWR@print@normalsize
\LWR@schemata@origSchema[#1]{#2}{#3}{#4}{#5}%
\end{lateximage}%
}
```

File 245 `lwarp-scrextend.sty`

§ 337 Package

scrextend

Pkg `scrextend` **scrextend** is emulated.

This package may be loaded standalone, but is also loaded automatically if **koma-script** classes are in use. `\DeclareDocumentCommand` is used to overwrite the **koma-script** definitions.

for HTML output:

```
\LWR@ProvidesPackageDrop{scrextend}

\DeclareDocumentCommand{\setkomafont}{m m}{}
\DeclareDocumentCommand{\addkomafont}{m m}{}
\DeclareDocumentCommand{\usekomafont}{m}{}

\DeclareDocumentCommand{\usefontofkomafont}{m}{}
\DeclareDocumentCommand{\useencodingofkomafont}{m}{}
\DeclareDocumentCommand{\usesizeofkomafont}{m}{}
\DeclareDocumentCommand{\usefamilyofkomafont}{m}{}
\DeclareDocumentCommand{\useseriesofkomafont}{m}{}
\DeclareDocumentCommand{\useshapeofkomafont}{m}{}

\AtBeginDocument{
\let\LWR@maketitle\maketitle
\DeclareDocumentCommand{\maketitle}{o}{\LWR@maketitle}
}

\DeclareDocumentCommand{\extratitle}{m}{}
\DeclareDocumentCommand{\titlehead}{m}{}
\DeclareDocumentCommand{\subject}{m}{}
\DeclareDocumentCommand{\publishers}{m}{\published{#1}}
\DeclareDocumentCommand{\uppertitleback}{m}{}
\DeclareDocumentCommand{\lowertitleback}{m}{}
\DeclareDocumentCommand{\dedication}{m}{}

\DeclareDocumentCommand{\ifthispageodd}{m m}{#1}

\DeclareDocumentCommand{\titlepagestyle}{}{}

\DeclareDocumentCommand{\cleardoublepageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleplainpage}{}{}
\DeclareDocumentCommand{\cleardoublestandardpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddpage}{}{}
```

```
\DeclareDocumentCommand{\cleardoubleoddpageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleoddemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleoddplainpage}{}{}
\DeclareDocumentCommand{\cleardoubleoddstandardpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenpageusingstyle}{m}{}
\DeclareDocumentCommand{\cleardoubleevenemptypage}{}{}
\DeclareDocumentCommand{\cleardoubleevenplainpage}{}{}
\DeclareDocumentCommand{\cleardoubleevenstandardpage}{}{}

\DeclareDocumentCommand{\multiplefootnoteseparator}{}{%
  \begingroup\let\thefootnotemark\multfootsep\@makefnmark\endgroup
}

\DeclareDocumentCommand{\multfootsep}{}{,}

\DeclareDocumentCommand{\footref}{m}{%
  \begingroup
    \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
  \endgroup
  \@footnotemark
}

\DeclareDocumentCommand{\deffootnote}{o m m m}{}
\DeclareDocumentCommand{\deffootnotemark}{m}{}
\DeclareDocumentCommand{\setfootnoterule}{o m}{}
\DeclareDocumentCommand{\raggedfootnote}{}{}

\DeclareDocumentCommand{\dictum}{o m}{
\begin{LWR@BlockClassWP}{\LWR@print@mbox{text-align:right}}{}{dictum}
    #2
    \IfValueT{#1}
    {
        \ifbool{FormatWP}
        {\begin{BlockClass}[\LWR@print@mbox{border-top:} 1px solid gray]{dictumauthor}}
        {\begin{BlockClass}{dictumauthor}}
        \dictumauthorformat{#1}
        \end{BlockClass}
    }
\end{LWR@BlockClassWP}
}

\DeclareDocumentCommand{\dictumwidth}{}{}
\DeclareDocumentCommand{\dictumauthorformat}{m}{(#1)}
\DeclareDocumentCommand{\dictumrule}{}{}
\DeclareDocumentCommand{\raggeddictum}{}{}
\DeclareDocumentCommand{\raggeddictumtext}{}{}
\DeclareDocumentCommand{\raggeddictumauthor}{}{}

\DeclareDocumentEnvironment{labeling}{o m}
```

```
{%
\def\sc@septext{#1}%
\list{}{}%
\let\makelabel\labelinglabel%
}
{
\endlist
}

\DeclareDocumentCommand{\labelinglabel}{m}{%
#1 \qquad \sc@septext%
}

\let\addmargin\relax
\let\endaddmargin\relax
\cslet{addmargin*}{\relax}
\cslet{endaddmargin*}{\relax}

\NewDocumentEnvironment{addmargin}{s O{} m}
{
\setlength{\LWR@templengthtwo}{#3}
\ifblank{#2}
{
    \begin{BlockClass}[
        \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthtwo}} ;
        \LWR@print@mbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
    ]{addmargin}
}
{
    \setlength{\LWR@templengthone}{#2}
    \begin{BlockClass}[
        \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ;
        \LWR@print@mbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
    ]{addmargin}
}
}
{\end{BlockClass}}
```

Ref to create a starred environment:
https://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment

```

\ExplSyntaxOn
\cs_new:cpn {addmargin*} {\addmargin*}
\cs_new_eq:cN {endaddmargin*} \endaddmargin
\ExplSyntaxOff

\DeclareDocumentCommand{\marginline}{m}{\marginpar{#1}}
```

File 246 `lwarp-scrhack.sty`

§ 338 Package scrhack

Pkg `scrhack` **scrhack** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{scrhack}
```

File 247 `lwarp-scrlayer.sty`

§ 339 Package scrlayer

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer` **scrlayer** is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{scrlayer}

\newcommand*{\DeclareSectionNumberDepth}[2]{}
\newcommand*{\DeclareLayer}[2][]{}
\newcommand*{\DeclareNewLayer}[2][]{}
\newcommand*{\ProvideLayer}[2][]{}
\newcommand*{\RedeclareLayer}[2][]{}
\newcommand*{\ModifyLayer}[2][]{}
\newcommand*{\layerhalign}{}
\newcommand*{\layervalign}{}
\newcommand*{\layerxoffset}{}
\newcommand*{\layeryoffset}{}
\newcommand*{\layerwidth}{}
\newcommand*{\layerheight}{}
\providecommand*{\LenToUnit}[1]{\strip@pt\dimexpr#1*\p@/\unitlength}
\newcommand*{\putUL}[1]{}
\newcommand*{\putUR}[1]{}
\newcommand*{\putLL}[1]{}
\newcommand*{\putLR}[1]{}
\newcommand*{\putC}[1]{}
\newcommand*{\GetLayerContents}[1]{}
\newcommand{\IfLayerExists}[3]{#3}
\newcommand*{\DestroyLayer}[1]{}
\newcommand*{\layercontentsmeasure}{}
```

```
\newcommand*{\currentpagestyle}{}
\newcommand*{\BeforeSelectAnyPageStyle}[1]{}
\newcommand*{\AfterSelectAnyPageStyle}[1]{}
\newcommand*{\DeclarePageStyleAlias}[2]{}
\newcommand*{\DeclareNewPageStyleAlias}[2]{}
\newcommand*{\ProvidePageStyleAlias}[2]{}
\newcommand*{\RedeclarePageStyleAlias}[2]{}
\newcommand*{\DestroyPageStyleAlias}[1]{}
\newcommand*{\GetRealPageStyle}[1]{}
\newcommand*{\DeclarePageStyleByLayers}[3][]{}
\newcommand*{\DeclareNewPageStyleByLayers}[3][]{}
\newcommand*{\ProvidePageStyleByLayers}[3][]{}
\newcommand*{\RedeclarePageStyleByLayers}[3][]{}
\NewDocumentCommand{\ForEachLayerOfPageStyle}{s m m}{}
\newcommand*{\AddLayersToPageStyle}[2]{}
\newcommand*{\AddLayersAtBeginOfPageStyle}[2]{}
\newcommand*{\AddLayersAtEndOfPageStyle}[2]{}
\newcommand*{\RemoveLayersFromPageStyle}[2]{}
\newcommand*{\AddLayersToPageStyleBeforeLayer}[3]{}
\newcommand*{\AddLayersToPageStyleAfterLayer}[3]{}
\newcommand*{\UnifyLayersAtPageStyle}[1]{}
\newcommand*{\ModifyLayerPageStyleOptions}[2]{}
\newcommand*{\AddToLayerPageStyleOptions}[2]{}
\newcommand{\IfLayerPageStyleExists}[3]{#3}
\newcommand{\IfRealLayerPageStyleExists}[3]{#3}
\newcommand{\IfLayerAtPageStyle}[4]{#4}
\newcommand{\IfSomeLayerAtPageStyle}[4]{#4}
\newcommand{\IfLayersAtPageStyle}[4]{#4}
\newcommand*{\DestroyRealLayerPageStyle}[1]{}
\@ifundefined{footheight}{\newlength\footheight}{}
\DeclareDocumentCommand{\automark}{s o m}{}
\DeclareDocumentCommand{\manualmark}{}{}
\DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
\DeclareDocumentCommand{\GenericMarkFormat}{}{}
\newcommand*{\@mkleft}[1]{}
\newcommand*{\@mkright}[1]{}
\newcommand*{\@mkdouble}[1]{}
\newcommand*{\@mkboth}[2]{}
\newcommand*{\scrlayerInitInterface}[1][]{}
\newcommand{\scrlayerAddToInterface}[3][]{}
\newcommand{\scrlayerAddCsToInterface}[3][]{}
\newcommand{\scrlayerOnAutoRemoveInterface}[2][]{}
```

File 248 `lwarp-scrlayer-notecolumn.sty`

§ 340 Package scrlayer-notecolumn

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer-notecolumn` **scrlayer-notecolumn** is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{scrlayer-notecolumn}

\newcommand*{\DeclareNoteColumn}[2][]{}
\newcommand*{\DeclareNewNoteColumn}[2][]{}
\newcommand*{\ProvideNoteColumn}[2][]{}
\newcommand*{\RedeclareNoteColumn}[2][]{}
\NewDocumentCommand{\makenote}{s o m}{\marginpar{#3}}
\newcommand*{\syncwithnotecolumn}[1][]{}
\newcommand*{\syncwithnotecolumns}[1][]{}
\newcommand*{\clearnotecolumn}[1][]{}
\newcommand*{\clearnotecolumns}[1][]{}
```

File 249 `lwarp-scrlayer-scrpage.sty`

§ 341 Package scrlayer-scrpage

(*Emulates or patches code by* Markus Kohm.)

Pkg `scrlayer-scrpage` **scrlayer-scrpage** is emulated.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{scrlayer-scrpage}

\@ifundefined{footheight}{\newlength\footheight}{}
\NewDocumentCommand{\lehead}{s o m}{}
\NewDocumentCommand{\cehead}{s o m}{}
\NewDocumentCommand{\rehead}{s o m}{}
\NewDocumentCommand{\lohead}{s o m}{}
\NewDocumentCommand{\cohead}{s o m}{}
\NewDocumentCommand{\rohead}{s o m}{}
\NewDocumentCommand{\lefoot}{s o m}{}
\NewDocumentCommand{\cefoot}{s o m}{}
```

```
\NewDocumentCommand{\refoot}{s o m}{}
\NewDocumentCommand{\lofoot}{s o m}{}
\NewDocumentCommand{\cofoot}{s o m}{}
\NewDocumentCommand{\rofoot}{s o m}{}
\NewDocumentCommand{\ohead}{s o m}{}
\NewDocumentCommand{\chead}{s o m}{}
\NewDocumentCommand{\ihead}{s o m}{}
\NewDocumentCommand{\ofoot}{s o m}{}
\NewDocumentCommand{\cfoot}{s o m}{}
\NewDocumentCommand{\ifoot}{s o m}{}
\DeclareDocumentCommand{\automark}{s o m}{}
\DeclareDocumentCommand{\manualmark}{}{}
\DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
\DeclareDocumentCommand{\GenericMarkFormat}{}{}
\newcommand*{\defpairofpagestyles}[3][]{}
\newcommand*{\newpairofpagestyles}[3][]{}
\newcommand*{\renewpairofpagestyles}[3][]{}
\newcommand*{\providepairofpagestyles}[3][]{}
\newcommand*{\clearmainofpairofpagestyles}
\newcommand*{\clearplainofpairofpagestyles}
\newcommand*{\clearpairofpagestyles}
\NewDocumentCommand{\deftriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\newtriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\renewtriplepagestyle}{m o o m m m m m m}{}
\NewDocumentCommand{\providetriplepagestyle}{m o o m m m m m m}{}
\newcommand*{\defpagestyle}[3]{}
\newcommand*{\newpagestyle}[3]{}
\newcommand*{\providepagestyle}[3]{}
\newcommand*{\renewpagestyle}[3]{}
```

File 250 **lwarp-section.sty**

§ 342 Package section

Pkg `section` **section** is ignored.

(*Emulates or patches code by* Oliver Pretzel.)

for HTML output:

```
\LWR@ProvidesPackageDrop{section}

\ifx\chapter\undefined
 \def\chsize{\Large}\def\hdsize{\huge}\else
 \def\chsize{\huge}\def\hdsize{\Huge}
\fi
\let\ttsize\LARGE
\let\ausize\large
\let\dasize\large
```

```
\let\secsize\Large
\let\subsize\large
\let\hdpos\raggedright
\newcounter{hddepth}
\let\fpind\relax
\def\ttfnt{}
\def\hdfnt{}
\def\fefnt{}
\def\thfnt{}
\def\pgfnt{}
\def\hmkfnt{}
\let\mkcse\uppercase
\def\hddot{}
\def\cpdot{:}
\def\nmdot{}
\ifx\secindent\undefined
 \newdimen\secindent
 \newskip\secpreskp
 \newskip\secpstskp
 \newdimen\subindent
 \newskip\subpreskp
 \newskip\subpstskp
 \newskip\parpstskp
 \newcount\c@hddepth
\fi
```

File 251 `lwarp-sectionbreak.sty`

§ 343 Package sectionbreak

(*Emulates or patches code by* Michal Hoftich.)

Pkg `sectionbreak` **sectionbreak** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{sectionbreak}

\renewcommand\asterism{\HTMLunicode{2042}}

\renewcommand\pre@sectionbreak{}
\renewcommand\post@sectionbreak{}

\renewcommand\print@sectionbreak[1]{%
\begin{center}
#1
\end{center}
}

```

File 252 `lwarp-sectsty.sty`

§ 344 Package sectsty

(*Emulates or patches code by* Rowland McDonnell.)

Pkg `sectsty` **sectsty** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{sectsty}

\newcommand*{\partfont}          [1] {}
\newcommand*{\partnumberfont}    [1] {}
\newcommand*{\parttitlefont}     [1] {}
\newcommand*{\chapterfont}       [1] {}
\newcommand*{\chapternumberfont} [1] {}
\newcommand*{\chaptertitlefont}  [1] {}
\newcommand*{\sectionfont}       [1] {}
\newcommand*{\subsectionfont}    [1] {}
\newcommand*{\subsubsectionfont} [1] {}
\newcommand*{\paragraphfont}     [1] {}
\newcommand*{\subparagraphfont}  [1] {}
\newcommand*{\minisecfont}  [1] {}
\newcommand*{\allsectionsfont}[1] {}
\newcommand{\nohang}{}
```

`\sectionrule` is only to be used in `*font` commands, thus it is ignored.

```
\newcommand*{\sectionrule}[5]{}

\def\ulemheading#1#2{}
```

File 253 `lwarp-setspace.sty`

§ 345 Package setspace

(*Emulates or patches code by* Robin Fairbairns.)

Pkg `setspace` **setspace** is not used during HTML conversion.

Discard all options for **lwarp-setspace**:

for HTML output:

```
\LWR@ProvidesPackageDrop{setspace}

\newcommand*{\setstretch}[1]{}
\newcommand*{\SetSinglespace}[1]{}
\newcommand*{\singlespacing}{}
\newcommand*{\onehalfspacing}{}
\newcommand*{\doublespacing}{}

\newenvironment*{singlespace}
{
\LWR@forcenewpage
\BlockClass{singlespace}
}
{\endBlockClass}

\newenvironment*{singlespace*}
{
\LWR@forcenewpage
\BlockClass{singlespace}
}
{\endBlockClass}

\newenvironment*{spacing}[1]{

}{

}

\newenvironment*{onehalfspace}
{
\LWR@forcenewpage
\BlockClass{onehalfspace}
}
{\endBlockClass}

\newenvironment*{doublespace}
{
\LWR@forcenewpage
\BlockClass{doublespace}
}
{\endBlockClass}
```

File 254 `lwarp-shadow.sty`

§ 346 Package shadow

(*Emulates or patches code by* Mauro Orlandini.)

Pkg shadow **shadow** is emulated.

for HTML output: Discard all options for **lwarp-shadow**:

```
\LWR@ProvidesPackageDrop{shadow}

\newdimen\sboxsep
\newdimen\sboxrule
\newdimen\sdim

\newcommand{\shabox}[1]{%
\InlineClass{shabox}{#1}%
}
```

File 255 `lwarp-showidx.sty`

§ 347 Package showidx

Pkg showidx **showidx** is ignored.

for HTML output: Discard all options for **lwarp-showidx**:

```
\LWR@ProvidesPackageDrop{showidx}
```

`\@wrindex` is redefined `\AtBeginDocument` by the **lwarp** core.

File 256 `lwarp-showkeys.sty`

§ 348 Package showkeys

(*Emulates or patches code by* David Carlisle, Morten Høgholm.)

Pkg showkeys **showkeys** is ignored.

for HTML output: Discard all options for **lwarp-showkeys**:

```
\LWR@ProvidesPackageDrop{showkeys}

\NewDocumentCommand{\showkeys}{s}{}
```

File 257 `lwarp-sidecap.sty`

§ 349 Package

sidecap

(*Emulates or patches code by* Rolf Niepraschk, Hubert Gässlein.)

Pkg `sidecap` **sidecap** is emulated.

for HTML output: Discard all options for **lwarp-sidecap**.

```
\LWR@ProvidesPackageDrop{sidecap}
```

See:
http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment
regarding the creation of starred environments with **xparse**.

```
\NewDocumentEnvironment{SCtable}{soo}
{\IfValueTF{#3}{\table[#3]}{\table}}
{\endtable}

\ExplSyntaxOn
\cs_new:cpn {SCtable*} {\SCtable*}
\cs_new_eq:cN {endSCtable*} \endSCtable
\ExplSyntaxOff


\NewDocumentEnvironment{SCfigure}{soo}
{\IfValueTF{#3}{\figure[#3]}{\figure}}
{\endfigure}

\ExplSyntaxOn
\cs_new:cpn {SCfigure*} {\SCfigure*}
\cs_new_eq:cN {endSCfigure*} \endSCfigure
\ExplSyntaxOff


\newenvironment*{wide}{}{}
```

File 258 `lwarp-sidenotes.sty`

§ 350 Package

sidenotes

(*Emulates or patches code by* Andy Thomas, Oliver Schebaum.)

Pkg `sidenotes` Patched for **lwarp**.

for HTML output: Load the original package:

```
\LWR@ProvidesPackagePass{sidenotes}
```

The following patch **sidenotes** for use with **lwarp**:

`\sidecaption` * [⟨*entry*⟩] [⟨*offset*⟩] {⟨*text*⟩}

```
\RenewDocumentCommand \sidecaption {s o o +m}
{
    \LWR@stoppars
    \begingroup
  \captionsetup{style=sidecaption}
  \IfBooleanTF{#1}
  { % starred
    \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
    \caption*{#4}
    \end{BlockClass}
  }
  { % unstarred
  \IfNoValueOrEmptyTF{#2}
    {\def\@sidenotes@sidecaption@tof{#4}}
    {\def\@sidenotes@sidecaption@tof{#2}}
    \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
    \caption[\@sidenotes@sidecaption@tof]{#4}
    \end{BlockClass}
  }
    \endgroup
    \LWR@startpars
}
```

Borrowed from the **lwarp** version of **keyfloat**:

```
\NewDocumentEnvironment{KFLTsidenotes@marginfloat}{O{-1.2ex} m}
{% start
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
```

```
}
{%
\endLWR@BlockClassWP%
}

\RenewDocumentEnvironment{marginfigure}{o}
  {\begin{KFLTsidenotes@marginfloat}{figure}}
  {\end{KFLTsidenotes@marginfloat}}

\RenewDocumentEnvironment{margintable}{o}
  {\begin{KFLTsidenotes@marginfloat}{table}}
  {\end{KFLTsidenotes@marginfloat}}
```

The following were changed by **sidenotes**, and now are reset back to their **lwarp**-supported originals:

Restoring the definition from the LaTeX 2ε `article.cls` source:

```
\renewenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}

\renewenvironment{table*}
               {\@dblfloat{table}}
               {\end@dblfloat}
```

File 259 `lwarp-SIunits.sty`

§ 351 Package

SIunits

(*Emulates or patches code by* Marcel Heldoorn.)

Pkg `SIunits` **SIunits** is patched for use by **lwarp**.

It is recommended to use `\unit` where possible, which combines the entire expression into a single `lateximage`, and adds the `alt` tag containing the LaTeX code, allowing for copy/paste. When units are used outside of the `\unit` macro, each unit macro will have its own `lateximage`, and each will have the `alt` tag set to "`(<\mathimagename>)`", which defaults to "`(math image)`".

for HTML output:

```
\LWR@ProvidesPackagePass{SIunits}
```

Patched for copy/paste with the HTML `alt` tag:

```
\DeclareRobustCommand{\unit}[2]{%
\@inunitcommandtrue%    original
```

```
\LWR@subsingledollar*%  lwarp
{% alt tag
    \textbackslash{}unit\{\LWR@HTMLsanitize{#1}\}%
        \{ \LWR@HTMLsanitize{#2}\}% extra space
}%
{SIunits}% add'l hashing
{%
    \LWR@origensuredmath{%  lwarp modification
        \SI@fstyle{#1\@qsk\period@active{#2}}%  original
    }%
}% contents
\@inunitcommandfalse%   original
}
```

File 260 **lwarp-siunitx.sty**

§ 352 Package

siunitx

(*Emulates or patches code by* Joseph Wright.)

Pkg siunitx **siunitx** is patched for use by **lwarp**.

fractions Due to **pdftolatex** limitations, fraction output is replaced by symbol output for `per-mode` and `quotient-mode`.

⚠ math mode required Some units will require that the expression be placed inside math mode.

NOTE: As of this writing, the **siunitx** extension for MathJax is not currently hosted at any public CDN, thus **siunitx** is not usable with MathJax unless a local copy of this extension is created first.

⚠ tabular Tabular `S` columns are rendered as simple `c` columns, and tabular `s` columns are not supported. These may be replaced by `c` columns with each cell contained in `\num` or `\si`.

for HTML output:

```
\RequirePackage{xcolor}% for \convertcolorspec

\LWR@ProvidesPackagePass{siunitx}


\AtBeginDocument{% in case textcomp was not loaded
\DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
\DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
\DeclareSIUnit\elementarycharge{\textit{e}}
\DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
\DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
\DeclareSIUnit\planckbar{\LWR@siunitx@textplanckbar}
}% AtBeginDocument
```

`\@ensuredmath` is not supported inside an `\hbox`, so it must temporarily be restored to its original. Similar for `\mbox`. SVG math is created explicitly when necessary, using `\LWR@subsingledollar`.

```
\ExplSyntaxOn
%
```

Modified to set set HTML `\textcolor` if not black:

```
\cs_undefine:N \__siunitx_print_aux:
\cs_new_protected:Npn \__siunitx_print_aux:
  {
    \text
      {
        \__siunitx_ensure_ltr:n
          {
            \color@begingroup
            \__siunitx_print_color:
            \__siunitx_font_shape:
            \__siunitx_font_weight:
            \use:c
              {
                @@_ \l__siunitx_print_type_tl _
                text \l__siunitx_font_family_tl :
              }
            \bool_if:NTF \l__siunitx_font_math_mode_bool
              { \__siunitx_print_math: }
              {
                \LWR@findcurrenttextcolor% lwarp
                \ifdefstring{\LWR@tempcolor}{000000}% lwarp
                {\__siunitx_print_text:}% lwarp
                {% lwarp
                    \LWR@textcurrentcolor{% lwarp
                        \__siunitx_print_text:
                    }% lwarp
                }% lwarp
              }
            \color@endgroup
          }
      }
  }


\cs_undefine:N \__siunitx_set_math_fam:n
\cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
  \int_new:c { c__siunitx_math #1 _int }
  \group_begin:% lwarp
    \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
```

```
    \LetLtxMacro\mbox\LWR@print@mbox% lwarp
    \hbox_set:Nn \l__siunitx_tmp_box
      {
        \ensuremath
          {
            \use:c { math #1 }
              {
                \int_gset:cn { c__siunitx_math #1 _int } { \fam }
              }
          }
      }
  \group_end:% lwarp
}

\cs_undefine:N \__siunitx_combined_output:n
\cs_new_protected:Npn \__siunitx_combined_output:n #1 {
    \group_begin:% lwarp
    \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
    \LetLtxMacro\mbox\LWR@print@mbox% lwarp
  \bool_if:NTF \l__siunitx_number_parse_bool
    {
      \tl_clear:N \l__siunitx_number_out_tl
      \bool_set_false:N \l__siunitx_number_compound_bool
      \__siunitx_number_output_parse:n {#1}
    }
    {
      \__siunitx_unit_output_pre_print:
```

For `parse-numbers=false`:

```
%       \__siunitx_print:nn { number } { \ensuremath {#1} }
        \LWR@subsingledollar{% lwarp
            \textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)% lwarp
        }{siunitx}{%
            \__siunitx_print:nn { number } {%
                \LWR@origensuredmath{#1}%
            }%
        }% lwarp

      \__siunitx_unit_output_print:
    }
    \group_end:% lwarp
}
%
```

For quotients, the fraction code is replaced by the symbol code:

```
\cs_undefine:N \__siunitx_number_output_quotient_fraction:
\cs_new_protected:Npn \__siunitx_number_output_quotient_fraction: {
```

```
  \bool_set_true:N \l__siunitx_number_compound_bool
  \__siunitx_number_output_quotient_aux_i:
  \tl_set_eq:NN \l__siunitx_number_out_tl
    \l__siunitx_number_numerator_tl
  \tl_put_right:NV \l__siunitx_number_out_tl \l__siunitx_output_quotient_tl
  \tl_put_right:NV \l__siunitx_number_out_tl
    \l__siunitx_number_denominator_tl
  \__siunitx_number_output_single_aux:
}
```

For units, the fraction code is replaced by the symbol code:

```
\cs_undefine:N \__siunitx_unit_format_fraction_fraction:
\cs_new_protected:Npn \__siunitx_unit_format_fraction_fraction: {
  \__siunitx_unit_format_fraction_symbol_aux:
  \int_compare:nNnT { \l__siunitx_unit_denominator_int } > { 1 }
    {
      \bool_if:NT \l__siunitx_unit_denominator_bracket_bool
        {
          \tl_put_left:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_open_tl
          \tl_put_right:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_close_tl
        }
    }
  \tl_set_eq:NN \l__siunitx_unit_tl \l__siunitx_unit_numerator_tl
  \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_per_symbol_tl
  \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_unit_denominator_tl
}
```

```
\cs_undefine:N \__siunitx_angle_print_astronomy_aux:
\cs_new_protected:Npn \__siunitx_angle_print_astronomy_aux: {
  \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-integer }
    \l__siunitx_tmpa_tl
    { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}% lwarp
 {% lateximage
  \hbox_set:Nn \l__siunitx_angle_marker_box
    {
      \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
    }
  \hbox_set:Nn \l__siunitx_angle_unit_box
    {
      \__siunitx_print:nV { unit } \l__siunitx_unit_tl
      \skip_horizontal:n { -\scriptspace }
    }
  \__siunitx_angle_print_astronomy_aux:n { marker }
  \__siunitx_angle_print_astronomy_aux:n { unit }
  \hbox_set:Nn \l__siunitx_angle_marker_box
    {
      \box_use:N \l__siunitx_angle_marker_box
```

```
      \box_use:N \l__siunitx_angle_unit_box
    }
  \dim_compare:nNnTF
    { \l__siunitx_angle_marker_dim } > { \l__siunitx_angle_unit_dim }
    { \__siunitx_angle_print_astronomy_marker: }
    { \__siunitx_angle_print_astronomy_unit: }
 }% lateximage
 {% not a lateximage
      \__siunitx_print:nV { unit } \l__siunitx_unit_tl
      \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
 }% not a lateximage
  \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-decimal }
    \l__siunitx_tmpa_tl
    { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
}

\RenewDocumentCommand \num { o m } {
  \leavevmode
  \group_begin:% lwarp
    \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
    \LetLtxMacro\mbox\LWR@print@mbox% lwarp
    \bool_set_false:N \l__siunitx_font_set_bool
    \IfNoValueF {#1}
      { \keys_set:nn { siunitx } {#1} }
    \__siunitx_number_output:n {#2}
  \group_end:% lwarp
}

\RenewDocumentCommand \numrange { o m m } {
  \leavevmode
  \group_begin:% lwarp
    \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
    \LetLtxMacro\mbox\LWR@print@mbox% lwarp
    \bool_set_false:N \l__siunitx_font_set_bool
    \IfNoValueF {#1}
      { \keys_set:nn { siunitx } {#1} }
    \__siunitx_range_numbers:nn {#2} {#3}
  \group_end:% lwarp
}

\RenewDocumentCommand \ang { o > { \SplitArgument { 2 } { ; } } m } {
  \group_begin:% lwarp
    \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
    \LetLtxMacro\mbox\LWR@print@mbox% lwarp
    \IfNoValueF {#1}
      { \keys_set:nn { siunitx } {#1} }
    \__siunitx_angle_output:nnn #2
  \group_end:% lwarp
}
```

```
189
190 \RenewDocumentCommand \si { o m } {
191   \leavevmode
192   \group_begin:% lwarp
193     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
194     \LetLtxMacro\mbox\LWR@print@mbox% lwarp
195     \bool_set_false:N \l__siunitx_font_set_bool
196     \IfNoValueTF {#1}
197       { \__siunitx_unit_output:nn {#2} { } }
198       {
199         \keys_set:nn { siunitx } {#1}
200         \__siunitx_unit_output:nn {#2} {#1}
201       }
202   \group_end:% lwarp
203 }
204
205
206 \RenewDocumentCommand{\SIrange}{o m m m}
207 {%
208   \leavevmode
209   \group_begin:% lwarp
210     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
211     \LetLtxMacro\mbox\LWR@print@mbox% lwarp
212     \bool_set_false:N \l__siunitx_font_set_bool
213     \IfNoValueTF {#1}
214       { \__siunitx_range_unit:nnnn {#4} { } {#2} {#3} }
215       {
216         \keys_set:nn { siunitx } {#1}
217         \__siunitx_range_unit:nnnn {#4} {#1} {#2} {#3}
218       }
219   \group_end:% lwarp
220 }
221
222 \ExplSyntaxOff
```

File 261 **lwarp-soul.sty**

§ 353 Package soul

(*Emulates or patches code by* Melchior FRANZ.)

Pkg `soul` Emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{soul}[2003/11/17]
2 \RequirePackage{xcolor}% for \convertcolorspec
```

Storage for the colors to use:

```
\newcommand*{\LWR@soululcolor}{}

\newcommand*{\LWR@soulstcolor}{}

% \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}
% \newcommand*{\LWR@soulhlcolor}{LWR@soulhlcolordefault}
\newcommand*{\LWR@soulhlcolor}{}
```

`\so` {⟨*text*⟩}

Basic markup with css:

```
\newcommand{\so}[1]{%
\LWR@HTMLtextstyle{letter-spacing:.2ex}{letterspacing}{#1}%
}
```

`\caps` {⟨*text*⟩}

```
\newcommand{\caps}[1]{%
\LWR@HTMLtextstyle%
    {font-variant:small-caps;letter-spacing:.1ex}%
    {capsspacing}{#1}%
}
```

`\LWR@soulcolor` {⟨*text*⟩} {⟨*color*⟩} {⟨*class*⟩} {⟨*colorstyle*⟩} {⟨*FormatWPstyle*⟩}

Add colors if not empty:

```
\newcommand{\LWR@soulcolor}[5]{%
\ifcsempty{#2}%
{\LWR@HTMLtextstyle{#5}{#3}{#1}}%
{%
    \convertcolorspec{named}{\@nameuse{#2}}{HTML}\LWR@tempcolor%
    \LWR@htmlspanclass[#5;#4:\LWR@origpound\LWR@tempcolor]{#3}{#1}%
}%
}

\newcommand{\ul}[1]{%
\LWR@soulcolor{#1}{LWR@soululcolor}{uline}{text-decoration-color}%
    {text-decoration:underline; text-decoration-skip: auto;}%
}

\newcommand{\st}[1]{
\LWR@soulcolor{#1}{LWR@soulstcolor}{sout}{text-decoration-color}%
    {text-decoration:line-through}%
}

\newcommand{\hl}[1]{
```

```
\LWR@soulcolor{#1}{LWR@soulhlcolor}{highlight}{background-color}%
    {background:\LWR@origpound{}F8E800}
}
```

Nullified:

```
\newcommand*{\soulaccent}[1]{}
\newcommand*{\soulregister}[2]{}
\newcommand{\sloppyword}[1]{#1}
\newcommand*{\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
\newcommand*{\resetso}{}
\newcommand*{\capsdef}[5]{}
\newcommand*{\capsreset}{}
\newcommand*{\capssave}[1]{}
\newcommand*{\capsselect}[1]{}
\newcommand*{\setul}[2]{}
\newcommand*{\resetul}{}
\newcommand*{\setuldepth}[1]{}
\newcommand*{\setuloverlap}[1]{}
```

Set colors:

```
\newcommand*{\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
\newcommand*{\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
\newcommand*{\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}
```

Long versions of the user-level macros:

```
\let\textso\so
\let\textul\ul
\let\texthl\hl
\let\textcaps\caps
```

File 262 `lwarp-soulpos.sty`

§ 354 Package soulpos

(*Emulates or patches code by* Javier Bezos.)

Pkg `soulpos` **soulpos** is emulated.

for HTML output:

```
\RequirePackage{soul}
\RequirePackage{soulutf8}
\LWR@ProvidesPackageDrop{soulpos}
```

```
\NewDocumentCommand{\ulposdef}{m o m}{}
```

```
5
6 \newdimen\ulwidth
7
8 \newcommand\ifulstarttype[1]{%
9 \expandafter\@secondoftwo%
10 }
11
12 \newcommand\ifulendtype[1]{%
13 \expandafter\@secondoftwo%
14 }
15
16 \newcommand{\ulstarttype}{0}
17 \newcommand{\ulendtype}{0}
18 \newcommand\ulpostolerance{0}%
```

File 263 `lwarp-soulutf8.sty`

§ 355 Package soulutf8

Pkg soulutf8 **soulutf8** is emulated.

lwarp's HTML output naturally supports UTF-8 encoding.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{soulutf8}
```

File 264 `lwarp-splitidx.sty`

§ 356 Package splitidx

(*Emulates or patches code by* MARKUS KOHM.)

Pkg splitidx **splitidx** is patched for use by **lwarp**.

If the `latexmk` option is selected for **lwarp**, **latexmk** will compile the document but will *not* compile the indexes. ***lwarpmk printindex*** and ***lwarpmk htmlindex*** will still be required.

⚠ `\thepage` When using `\AtWriteToIndex` or `\AtNextWriteToIndex`, the user must not refer to `\thepage` during HTML output, as the concept of a page number is meaningless. Instead, do

```
\addtocounter{LWR@autoindex}{1}
\LWR@new@label{LWRindex-\arabic{LWR@autoindex}}
```

where the \index-like action occurs, and then refer to \arabic{LWR@autoindex} instead of \thepage where the reference should occur.

See section 422.17 in the **lwarp-patch-memoir** package for the \@@wrspindexhyp macro as an example.

for HTML output:

```
\LWR@ProvidesPackagePass{splitidx}

\catcode`\_=12%
\xpatchcmd{\newindex}
    {\jobname-#2.idx}
    {\jobname-#2_html.idx}
    {}
    {\LWR@patcherror{splitidx}{@newindex}}
\catcode`\_=8%
```

Patched to use **lwarp**'s automatic indexing counter instead of \thepage:

```
\renewcommand*{\@wrsindex}[2][]{%
  \ifx\relax#1\relax
    \if@splitidx
      \@wrsindex[idx]{#2}%
    \else
      \def\@tempa{#2}%
      \if@verbindex\@onelevel@sanitize\@tempa\fi
      \@wrindex{\@tempa}%
    \fi
  \else
    \def\@tempa{#2}%
    \csname index@#1@hook\endcsname
%     \expandafter\ifx\csname @@wrsindex\endcsname\relax
    \addtocounter{LWR@autoindex}{1}%                    lwarp
    \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%    lwarp
%       \@@@wrsindex{#1}{{\@tempa}{\thepage}}%
      \@@@wrsindex{#1}{{\@tempa}{\arabic{LWR@autoindex}}}%
%     \else
%       \def\@tempb{\@@wrsindex{#1}}%
%       \expandafter\@tempb\@tempa||\\%
%     \fi
    \endgroup
    \@esphack
  \fi
}
```

lwarp defines sectioning commands with **xparse**, so the below patches are done as temporary redefinitions instead of being \let.

```
\xpatchcmd{\printsubindex}
    {\let\section\subsection}
```

```
    {\renewcommand*{\section}{\subsection}}
    {}
    {\LWR@patcherror{splitidx}{printsubindex-section}}

\xpatchcmd{\printsubindex}
    {\let\chapter\section}
    {\renewcommand*{\chapter}{\section}}
    {}
    {\LWR@patcherror{splitidx}{printsubindex-chapter}}

\xpatchcmd{\printsubindex}
    {\let\@makechapterhead\section}
    {\def\@makechapterhead{\section}}
    {}
    {\LWR@patcherror{splitidx}{printsubindex-chapter}}
```

File 265 `lwarp-stabular.sty`

§ 357 Package stabular

(*Emulates or patches code by* Sigitas Tolušis.)

Pkg `stabular` **stabular** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{stabular}
```

Env `stabular` `[`⟨*vpos*⟩`]` `{`⟨*colspec*⟩`}`

```
\newenvironment{stabular}[2][c]
{
\begin{tabular}[#1]{#2}
\renewcommand{\noalign}[1]{}
}
{\end{tabular}}
```

Env `stabular` `{`⟨*width*⟩`}` `[`⟨*vpos*⟩`]` `{`⟨*colspec*⟩`}`

```
\NewDocumentEnvironment{stabular*}{m o m}
{
\begin{tabular}[#2]{#3}
\renewcommand{\noalign}[1]{}
}
{\end{tabular}}
```

File 266 `lwarp-stfloats.sty`

§ 358 Package stfloats

Pkg `stfloats` **stfloats** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{stfloats}

\newcommand*{\fnbelowfloat}{}
\newcommand*{\fnunderfloat}{}
\newcommand*{\setbaselinefloat}{}
\newcommand*{\setbaselinefixed}{}
```

File 267 `lwarp-subfig.sty`

§ 359 Package subfig

(*Emulates or patches code by* Steven Douglas Cochran.)

Pkg `subfig` **subfig** is supported and patched by **lwarp**.

⚠ `lof/lotdepth` At present, the package options for `lofdepth` and `lotdepth` are not working. These counters must be set separately after the package has been loaded.

horizontal spacing In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

for HTML output: Accept all options for **lwarp-subfig**:

```
\LWR@ProvidesPackagePass{subfig}
```

`\sf@@@subfloat` `{`⟨*1 type*⟩`}` `[`⟨*2 lof entry*⟩`]` `[`⟨*3 caption*⟩`]` `{`⟨*4 contents*⟩`}`

The outer minipage allows side-by-side subfloats with `\hfill` between.

```
\long\def\sf@@@subfloat#1[#2][#3]#4{%
\begin{minipage}{\linewidth}% lwarp

\IfValueTF{#2}{%
    \LWR@setlatestname{#2}%
}{%
```

```
    \IfValueTF{#3}{%
        \LWR@setlatestname{#3}%
    }{}%
}%
\LWR@stoppars% lwarp
    \@ifundefined{FBsc@max}{}%
        {\FB@readaux{\let\FBsuboheight\relax}}%
    \@tempcnta=\@ne
    \if@minipage
      \@tempcnta=\z@
    \else\ifdim \lastskip=\z@ \else
      \@tempcnta=\tw@
    \fi\fi
    \ifmaincaptiontop
      \sf@top=\sf@nearskip
      \sf@bottom=\sf@farskip
    \else
      \sf@top=\sf@farskip
      \sf@bottom=\sf@nearskip
    \fi
    \leavevmode

%     \setbox\@tempboxa \hbox{#4}%
%     \@tempdima=\wd\@tempboxa
%     \@ifundefined{FBsc@max}{}%
%         {\global\advance\Xhsize-\wd\@tempboxa
%          \dimen@=\ht\@tempboxa
%          \advance\dimen@\dp\@tempboxa
%          \ifdim\dimen@>\FBso@max
%            \global\FBso@max\dimen@
%          \fi}%
```

Do not use boxes, which interfere with `lateximages`:

```
%     \vtop%
    \bgroup
%       \vbox%
      \bgroup
        \ifcase\@tempcnta
          \@minipagefalse
        \or
%           \vskip\sf@top
        \or
          \ifdim \lastskip=\z@ \else
%             \@tempskipb\sf@top\relax\@xaddvskip
          \fi
        \fi
        \sf@ifpositiontop{%
          \ifx \@empty#3\relax \else
            \sf@subcaption{#1}{#2}{#3}%
```

```
%             \vskip\sf@capskip
%             \vskip\sf@captopadj
          \fi\egroup
%           \hrule width0pt height0pt depth0pt
            \LWR@startpars% lwarp
%  \box\@tempboxa
            #4
            \LWR@stoppars% lwarp
        }{%
        \LWR@startpars% lwarp
        \@ifundefined{FBsc@max}%
            {
%  \box\@tempboxa
                #4
            }%
            {\ifx\FBsuboheight\relax
%                \box\@tempboxa
                #4
             \else
%                \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
                #4
             \fi}%
        \LWR@stoppars% lwarp
          \egroup
          \ifx \@empty#3\relax \else
%             \vskip\sf@capskip
%             \hrule width0pt height0pt depth0pt
            \sf@subcaption{#1}{#2}{#3}%
         \fi
        }%
%       \vskip\sf@bottom
    \egroup
    \@ifundefined{FBsc@max}{}%
        {\addtocounter{FRobj}{-1}%
         \ifnum\c@FRobj=0\else
           \subfloatrowsep
         \fi}%
    \ifmaincaptiontop\else
      \global\advance\@nameuse{c@\@captype}\m@ne
    \fi
\end{minipage}% lwarp
\LWR@startpars% lwarp
  \endgroup\ignorespaces%
}%
```

`\sf@subcaption` {⟨*1 type*⟩} {⟨*2 lof entry*⟩} {⟨*3 caption*⟩}

```
\long\def\sf@subcaption#1#2#3{%
\LWR@stoppars% lwarp
```

```
\ifx \relax#2\relax \else
  \bgroup
    \let\label=\@gobble
    \let\protect=\string
    \def\@subcaplabel{%
      \caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
    \sf@updatecaptionlist{#1}{#2}{\the\value{\@captype}}{\the\value{#1}}%
  \egroup
\fi
\bgroup
  \ifx \relax#3\relax
    \let\captionlabelsep=\relax
  \fi
%     \setbox0\vbox{%
%        \hb@xt@\the\@tempdima{%
%
% %          \hss
% %         \parbox[t]{\the\@tempdima}{%
%           \caption@make
%               {\@nameuse{sub\@captype name}}%
%               {\@nameuse{thesub\@captype}}%
%               {#3}
% %  }%
% %            \hss
%   }
%  }%
   \@ifundefined{FBsc@max}%
%         {\box0}%
        {
%  \parbox[t]{\the\@tempdima}{%
\LWR@traceinfo{sfsubcap B1}% lwarp
            \LWR@figcaption% lwarp
            \caption@make
                {\@nameuse{sub\@captype name}}%
                {\@nameuse{thesub\@captype}}%
                {#3}

            \endLWR@figcaption% lwarp
\LWR@traceinfo{sfsubcap B2}% lwarp
%  }%
        }%
        {\dimen@\ht0%
         \advance\dimen@\dp0%
         \ifdim\dimen@>\FBsc@max
           \global\FBsc@max\dimen@
         \fi
         \FB@readaux{\let\FBsubcheight\relax}%
         \ifx\FBsubcheight\relax
           \def\next{
%  \parbox[t]{\the\@tempdima}
```

```
            }%
          \else
            \def\next{
%  \parbox[t][\FBsubcheight][t]{\the\@tempdima}
            }%
          \fi
%           \vbox{%
%             \hb@xt@\the\@tempdima{%

%               \hss
%               \next{%
\LWR@traceinfo{sfsubcap C1}% lwarp
                \caption@make
                    {\@nameuse{sub\@captype name}}%
                    {\@nameuse{thesub\@captype}}%
                    {#3}
\LWR@traceinfo{sfsubcap C1}% lwarp
%  }%
%               \hss

%  }
%           }
        }%
  \egroup
\LWR@startpars% lwarp
}
```

\subfloat@label Patches for \sf@sub@label:

```
\def\subfloat@label{%
\LWR@ensuredoingapar% lwarp
  \@ifnextchar(%   %) match left parenthesis
    {\sf@sub@label}
    {\sf@sub@label(Sub\@captype\space
                    \@ifundefined{thechapter}{}{\@nameuse{thechapter}\space}%
                    \@nameuse{p@sub\@captype}%
                    \@nameuse{thesub\@captype}.)}}
```

Patches for \subref.

\sf@subref {⟨*label*⟩}

The unstarred version uses a \ref link whose printed text comes from the sub@<label>:

```
\renewcommand{\sf@subref}[1]{%
\LWR@subnewref{#1}{sub@#1}%
}
```

\sf@@subref {⟨*label*⟩}

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```
\renewcommand{\sf@@subref}[1]{\LWR@orig@pageref{sub@#1}}
```

Defining new subfloats. The `l@sub<type>` for each is redefined.

`\@newsubfloat` `[`⟨*keys/values*⟩`]` `{`⟨*float name*⟩`}`

```
\LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat

\def\@newsubfloat[#1]#2{%
\LWR@orig@newsubfloat[#1]{#2}%
\renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}%
}
```

Pre-defined for figures and tables:

`\l@subfigure` `{`⟨*text*⟩`}` `{`⟨*pagenum*⟩`}`

```
\renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
```

`\l@subtable` `{`⟨*text*⟩`}` `{`⟨*pagenum*⟩`}`

```
\renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
```

File 268 `lwarp-subfigure.sty`

§ 360 Package

subfigure

Pkg `subfigure` **subfigure** is emulated by **subfig**.

for HTML output:

```
\LWR@ProvidesPackageDrop{subfigure}
\RequirePackage{subfig}

\LetLtxMacro\subfigure\subfloat
\LetLtxMacro\subtable\subfloat
\LetLtxMacro\Subref\subref
\@ifundefined{figuretopcaptrue}{\newif\iffiguretopcap}{}
\newif\ifsubfiguretopcap
\newif\ifsubcaphang
\newif\ifsubcapcenter
\newif\ifsubcapcenterlast
\newif\ifsubcapnooneline
\newif\ifsubcapraggedright
\newskip\subfigtopskip
\newskip\subfigcapskip
```

```
\newdimen\subfigcaptopadj
\newskip\subfigbottomskip
\newdimen\subfigcapmargin
\newskip\subfiglabelskip
\newcommand*{\subcapsize}{}
\newcommand*{\subcaplabelfont}{}
\newcommand*{\subcapfont}{}
```

File 269 `lwarp-supertabular.sty`

§ 361 Package

supertabular

(*Emulates or patches code by* Johannes Braams, Theo Jurriens.)

Pkg supertabular **supertabular** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{supertabular}
```

⚠ misplaced alignment
alignment tab character &

For `\tablefirsthead`, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\StopDefiningTabulars
```

See section 9.9.

⚠ latexmage **supertabular** and **xtab** are not supported inside a `latexmage`.

```
\newcommand{\LWRST@firsthead}{}

\newcommand{\tablefirsthead}[1]{%
    \long\gdef\LWRST@firsthead{#1}%
}

\newcommand{\tablehead}[1]{}
\newcommand{\tabletail}[1]{}

\newcommand{\LWRST@lasttail}{}

\newcommand{\tablelasttail}[1]{%
    \long\gdef\LWRST@lasttail{#1}%
}

\newcommand{\tablecaption}[2][]{%
    \long\gdef\LWRST@caption{\caption[#1]{#2}}%
}
```

```
\let\topcaption\tablecaption
\let\bottomcaption\tablecaption


\newcommand*{\LWRST@caption}{}

\newcommand*{\shrinkheight}[1]{}

\NewDocumentEnvironment{supertabular}{s o m}
{%
\LWR@traceinfo{supertabular}%
\table%
\LWRST@caption%
\begin{tabular}{#3}%
\TabularMacro\ifdefvoid{\LWRST@firsthead}%
{\LWR@getmynexttoken}%
{\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
}%
{%
\ifdefvoid{\LWRST@lasttail}%
{}%
{%
\TabularMacro\ResumeTabular%
\LWRST@lasttail%
}%
\end{tabular}%
\endtable%
\LWR@traceinfo{supertabular done}%
}

\NewDocumentEnvironment{mpsupertabular}{s o m}
{\minipage{\linewidth}\supertabular{#3}}
{\endsupertabular\endminipage}
```

File 270 `lwarp-syntonly.sty`

§ 362 Package syntonly

(*Emulates or patches code by* Frank Mittelbach, Rainer Schöpf.)

Pkg `syntonly` Emulated.

for HTML output: Discard all options for **lwarp-syntonly**:

```
\LWR@ProvidesPackageDrop{syntonly}
```

```
\newif\ifsyntax@
\syntax@false

\newcommand*{\syntaxonly}{}

\@onlypreamble\syntaxonly
```

File 271 `lwarp-t1enc.sty`

§ 363 Package t1enc

Pkg t1enc **t1enc** does not work with **lwarp**.

for HTML output:

```
\LWR@loadnever{t1enc}{fontenc, inputenc, inputenx}
```

File 272 `lwarp-tabls.sty`

§ 364 Package tabls

(*Emulates or patches code by* Donald Arseneau.)

Pkg tabls **tabls** is emulated. `\LWR@hline` is used to handle the optional argument when **tabls** is loaded.

for HTML output:

```
\LWR@ProvidesPackageDrop{tabls}

\newdimen\tablinesep
\newdimen\arraylinesep
\newdimen\extrarulesep
```

File 273 `lwarp-tabularx.sty`

§ 365 Package tabularx

(*Emulates or patches code by* David Carlisle.)

Pkg tabularx **tabularx** is emulated by **lwarp**.

for HTML output: Discard all options for **lwarp-tabularx**:

```
\LWR@ProvidesPackageDrop{tabularx}
```

```
\DeclareDocumentEnvironment{tabularx}{m o m}
{\tabular{#3}}
{\endtabular}

\DeclareDocumentEnvironment{tabularx*}{m o m}
{\tabular{#3}}
{\endtabular}
```

File 274 `lwarp-tabulary.sty`

§ 366 Package tabulary

(*Emulates or patches code by* David Carlisle.)

Pkg `tabulary` **tabulary** is emulated by **lwarp**.

for HTML output: Discard all options for **lwarp-tabulary**.

Column types `L`, `C`, `R`, and `J` are emulated by **lwarp** core code.

```
\LWR@ProvidesPackageDrop{tabulary}

\NewDocumentEnvironment{tabulary}{m o m}
{\tabular{#3}}
{\endtabular}

\NewDocumentEnvironment{tabulary*}{m o m}
{\tabular{#3}}
{\endtabular}

\newdimen\tymin
\newdimen\tymax
\def\tyformat{}
```

File 275 `lwarp-textarea.sty`

§ 367 Package textarea

(*Emulates or patches code by* Alexander I. Rozhenko.)

Pkg `textarea` **textarea** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{textarea}
```

```
\newcommand\StartFromTextArea{}
\newcommand\StartFromHeaderArea{}
\newcommand*\RestoreTextArea{}
\newcommand*\ExpandTextArea[1][*]{}
\let\NCC@restoretextarea\@empty
```

File 276 `lwarp-textcomp.sty`

§ 368 Package textcomp

(*Emulates or patches code by* Frank Mittelbach, Robin Fairbairns, Werner Lemberg.)

Pkg `textcomp` **textcomp** is patched for use by **lwarp**.

§ 368.1 Limitations

Some **textcomp** symbols do not have Unicode equivalents, and thus are not supported.

⚠ missing symbols Many **textcomp** symbols are not supported by many fonts. Try using more complete fonts in the css, but expect to see gaps in coverage.

§ 368.2 Package loading

for HTML output:

```
\LWR@ProvidesPackagePass{textcomp}
```

§ 368.3 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for LuaLaTeX and XeLaTeX.

§ 368.3.1 pdfLaTeX symbols

```
\AtBeginDocument{
\ifPDFTeX
\newcommand*{\LWR@HTML@textdegree}{\HTMLentity{deg}}
\newcommand*{\LWR@HTML@textcelsius}{\HTMLunicode{2103}}
\newcommand*{\LWR@HTML@textohm}{\HTMLunicode{2126}}
\newcommand*{\LWR@HTML@textmu}{\HTMLunicode{00B5}}
\newcommand*{\LWR@HTML@textlquill}{\HTMLunicode{2045}}
\newcommand*{\LWR@HTML@textrquill}{\HTMLunicode{2046}}
\newcommand*{\LWR@HTML@textcircledP}{\HTMLunicode{2117}}
\newcommand*{\LWR@HTML@texttwelveudash}{\HTMLunicode{2014}}% emdash
\newcommand*{\LWR@HTML@textthreequartersemdash}{\HTMLunicode{2014}}% emdash
```

```
\newcommand*{\LWR@HTML@textmho}{\HTMLunicode{2127}}
\newcommand*{\LWR@HTML@textnaira}{\HTMLunicode{20A6}}
\newcommand*{\LWR@HTML@textpeso}{\HTMLunicode{20B1}}
\newcommand*{\LWR@HTML@textrecipe}{\HTMLunicode{211E}}
\newcommand*{\LWR@HTML@textinterrobangdown}{\HTMLunicode{2E18}}
\newcommand*{\LWR@HTML@textperthousand}{\HTMLunicode{2030}}
\newcommand*{\LWR@HTML@textpertenthousand}{\HTMLunicode{2031}}
\newcommand*{\LWR@HTML@textbaht}{\HTMLunicode{0E3F}}
\newcommand*{\LWR@HTML@textdiscount}{\%}
\newcommand*{\LWR@HTML@textservicemark}{\HTMLunicode{2120}}
\else
```

§ 368.3.2 XeLaTeX and LuaLaTeX symbols

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```
\newcommand*{\LWR@HTML@textdegree}{°}
\newcommand*{\LWR@HTML@textcelsius}{℃}
\newcommand*{\LWR@HTML@textohm}{Ω}
\newcommand*{\LWR@HTML@textmu}{µ}
\newcommand*{\LWR@HTML@textlquill}{⁅}
\newcommand*{\LWR@HTML@textrquill}{⁆}
\newcommand*{\LWR@HTML@textcircledP}{℗}
\newcommand*{\LWR@HTML@texttwelveudash}{--}% emdash
\newcommand*{\LWR@HTML@textthreequartersemdash}{--}% emdash
\newcommand*{\LWR@HTML@textmho}{℧}
\newcommand*{\LWR@HTML@textnaira}{₦}
\newcommand*{\LWR@HTML@textpeso}{₱}
\newcommand*{\LWR@HTML@textrecipe}{℞}
\newcommand*{\LWR@HTML@textinterrobangdown}{¡}
\newcommand*{\LWR@HTML@textperthousand}{‰}
\newcommand*{\LWR@HTML@textpertenthousand}{‱}
\newcommand*{\LWR@HTML@textbaht}{฿}
\newcommand*{\LWR@HTML@textdiscount}{\%}
\newcommand*{\LWR@HTML@textservicemark}{}
\fi

\LWR@formatted{textdegree}
\LWR@formatted{textcelsius}
\LWR@formatted{textohm}
\LWR@formatted{textmu}
\LWR@formatted{textlquill}
\LWR@formatted{textrquill}
\LWR@formatted{textcircledP}
\LWR@formatted{texttwelveudash}
\LWR@formatted{textthreequartersemdash}
\LWR@formatted{textmho}
\LWR@formatted{textnaira}
```

```
\LWR@formatted{textpeso}
\LWR@formatted{textrecipe}
\LWR@formatted{textinterrobangdown}
\LWR@formatted{textperthousand}
\LWR@formatted{textpertenthousand}
\LWR@formatted{textbaht}
\LWR@formatted{textdiscount}
\LWR@formatted{textservicemark}
```

§ 368.4 HTML dicritics

For HTML, Unicode diacritical marks are used:

```
\newcommand*{\LWR@HTML@capitalcedilla}[1]{#1\HTMLunicode{0327}}
\newcommand*{\LWR@HTML@capitalogonek}[1]{#1\HTMLunicode{0328}}
\newcommand*{\LWR@HTML@capitalgrave}[1]{#1\HTMLunicode{0300}}
\newcommand*{\LWR@HTML@capitalacute}[1]{#1\HTMLunicode{0301}}
\newcommand*{\LWR@HTML@capitalcircumflex}[1]{#1\HTMLunicode{0302}}
\newcommand*{\LWR@HTML@capitaltilde}[1]{#1\HTMLunicode{0303}}
\newcommand*{\LWR@HTML@capitaldieresis}[1]{#1\HTMLunicode{0308}}
\newcommand*{\LWR@HTML@capitalhungarumlaut}[1]{#1\HTMLunicode{30B}}
\newcommand*{\LWR@HTML@capitalring}[1]{#1\HTMLunicode{30A}}
\newcommand*{\LWR@HTML@capitalcaron}[1]{#1\HTMLunicode{30C}}
\newcommand*{\LWR@HTML@capitalbreve}[1]{#1\HTMLunicode{306}}
\newcommand*{\LWR@HTML@capitalmacron}[1]{#1\HTMLunicode{304}}
\newcommand*{\LWR@HTML@capitaldotaccent}[1]{#1\HTMLunicode{307}}
```

`\textcircled` becomes a span with a rounded border:

```
\newcommand*{\LWR@HTML@textcircled}[1]{%
\InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{#1}%
}

\LWR@formatted{capitalcedilla}
\LWR@formatted{capitalogonek}
\LWR@formatted{capitalgrave}
\LWR@formatted{capitalacute}
\LWR@formatted{capitalcircumflex}
\LWR@formatted{capitaltilde}
\LWR@formatted{capitaldieresis}
\LWR@formatted{capitalhungarumlaut}
\LWR@formatted{capitalring}
\LWR@formatted{capitalcaron}
\LWR@formatted{capitalbreve}
\LWR@formatted{capitalmacron}
\LWR@formatted{capitaldotaccent}
\LWR@formatted{textcircled}

}% AtBeginDocument
```

File 277 `lwarp-textfit.sty`

§ 369 Package textfit

Pkg `textfit` **textfit** is emulated.

Text is placed into a `<span>` of class `textfit`. Sizes are approximated, and also limited by browser min/max font-size settings.

for HTML output:

```
\LWR@ProvidesPackageDrop{textfit}

\newsavebox{\LWR@textfitbox}

\newcommand*{\LWR@textfitscale}[2]{%
\setlength{\LWR@templengthone}{#1}%
\setlength{\LWR@templengthone}{%
    1em*\ratio{\LWR@templengthone}{\LWR@templengththwo}%
}%
\InlineClass[font-size:\LWR@printlength{\LWR@templengthone}]{textfit}{#2}%
}

\newcommand*{\scaletowidth}[2]{%
\sbox{\LWR@textfitbox}{#2}%
\settowidth{\LWR@templengththwo}{\usebox{\LWR@textfitbox}}%
\LWR@textfitscale{#1}{#2}%
}

\newcommand*{\scaletoheight}[2]{%
\sbox{\LWR@textfitbox}{#2}%
\settoheight{\LWR@templengththwo}{\usebox{\LWR@textfitbox}}%
\LWR@textfitscale{#1}{#2}%
}
```

File 278 `lwarp-textpos.sty`

§ 370 Package textpos

(*Emulates or patches code by* Norman Gray.)

Pkg `textpos` **textpos** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{textpos}
```

```
\NewDocumentEnvironment{textblock}{m r()}{}{}
\NewDocumentEnvironment{textblock*}{m o r()}{}{}
\newcommand*{\TPGrid}[3][]{}
\NewDocumentCommand{\TPMargin}{s o}{}
\newcommand*{\textblockcolour}[1]{}
\newcommand*{\textblockrulecolour}[1]{}
\newcommand*{\textblockcolor}[1]{}
\newcommand*{\textblockrulecolor}[1]{}
\newcommand*{\tekstblokkulur}[1]{}
\newcommand*{\tekstblokrulekulur}[1]{}
\newlength{\TPHorizModule}
\newlength{\TPVertModule}
\newlength{\TPboxrulesize}
\newcommand{\textblocklabel}[1]{}
\newcommand*{\showtextsize}{}
\newcommand{\textblockorigin}[2]{}
```

File 279 `lwarp-theorem.sty`

§ 371 Package theorem

(*Emulates or patches code by* Frank Mittelbach.)

Pkg `theorem` **theorem** is patched for use by **lwarp**.

Table 14: Theorem package — CSS styling of theorems and proofs

Theorem: `<div>` of class `theorembody<theoremstyle>`

Theorem Header: `<span>` of class `theoremheader`

where `<theoremstyle>` is `plain`, `break`, etc.

for HTML output:

```
\LWR@ProvidesPackagePass{theorem}
```

§ 371.1 Remembering the theorem style

Storage for the style being used for new theorems:

```
\newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
\gdef\theoremstyle#1{%
   \@ifundefined{th@#1}{\@warning
```

```
          {Unknown theoremstyle ‘#1’. Using ‘plain’}%
          \theorem@style{plain}%
           \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
          }%
      {%
          \theorem@style{#1}%
          \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
      }%
      \begingroup
        \csname th@\the\theorem@style \endcsname
      \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
\gdef\@xnthm#1#2[#3]{%
   \expandafter\@ifdefinable\csname #1\endcsname
   {%
    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
    \@definecounter{#1}\@newctr{#1}[#3]%
    \expandafter\xdef\csname the#1\endcsname
      {\expandafter \noexpand \csname the#3\endcsname
       \@thmcountersep \@thmcounter{#1}}%
    \def\@tempa{\global\@namedef{#1}}%
    \expandafter \@tempa \expandafter{%
      \csname th@\the \theorem@style
            \expandafter \endcsname \the \theorem@bodyfont
     \@thm{#1}{#2}}%
    \global \expandafter \let \csname end#1\endcsname \@endtheorem
    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
   }}

\gdef\@ynthm#1#2{%
    \expandafter\@ifdefinable\csname #1\endcsname
   {
    \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
    \@definecounter{#1}%
    \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
    \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
     \expandafter{\csname th@\the \theorem@style \expandafter
     \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
    \global \expandafter \let \csname end#1\endcsname \@endtheorem
    \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
   }}

\gdef\@othm#1[#2]#3{%
  \expandafter\ifx\csname c@#2\endcsname\relax
   \@nocounterr{#2}%
  \else
```

```
  \expandafter\@ifdefinable\csname #1\endcsname
  {
   \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
   \expandafter \xdef \csname the#1\endcsname
    {\expandafter \noexpand \csname the#2\endcsname}%
   \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
    \expandafter{\csname th@\the \theorem@style \expandafter
    \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
   \global \expandafter \let \csname end#1\endcsname \@endtheorem
   \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\@nameuse{LWR@thmstyle#1}}}% lwarp
   }%
 \fi}
```

§ 371.2 CSS patches

The following are patched for CSS.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader`.

```
\gdef\th@plain{%
  \def\@begintheorem##1##2{%
        \item[
            \InlineClass{theoremheader}{##1\ ##2}
        ]
    }%
\def\@opargbegintheorem##1##2##3{%
   \item[
        \InlineClass{theoremheader}{##1\ ##2\ (##3)}
    ]
    }
}

\gdef\th@break{%
  \def\@begintheorem##1##2{%
    \item[
        \InlineClass{theoremheader}{##1\ ##2}\newline%
    ]
    }%
\def\@opargbegintheorem##1##2##3{%
  \item[
        \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
    ]
    }
}
```

```
87
88 \gdef\th@marginbreak{%
89   \def\@begintheorem##1##2{
90     \item[
91         \InlineClass{theoremheader}{##2 \qquad ##1}\newline
92     ]
93     }%
94 \def\@opargbegintheorem##1##2##3{%
95     \item[
96         \InlineClass{theoremheader}{##2 \qquad ##1\ %
97         (##3)}\newline
98     ]
99     }
100 }
101
102 \gdef\th@changebreak{%
103   \def\@begintheorem##1##2{
104     \item[
105         \InlineClass{theoremheader}{##2\ ##1}\newline
106     ]
107     }%
108 \def\@opargbegintheorem##1##2##3{%
109     \item[
110         \InlineClass{theoremheader}{ ##2\ ##1\ %
111         (##3)}\newline
112     ]
113     }
114 }
115
116 \gdef\th@change{%
117   \def\@begintheorem##1##2{
118     \item[
119         \InlineClass{theoremheader}{##2\ ##1}
120     ]
121     }%
122 \def\@opargbegintheorem##1##2##3{%
123     \item[
124         \InlineClass{theoremheader}{##2\ ##1\ (##3)}
125     ]
126     }
127 }
128
129 \gdef\th@margin{%
130   \def\@begintheorem##1##2{
131     \item[
132         \InlineClass{theoremheader}{##2 \qquad ##1}
133     ]
134     }%
135 \def\@opargbegintheorem##1##2##3{%
136     \item[
```

```
            \InlineClass{theoremheader}{##2 \qquad ##1\ (##3)}
    ]
    }
}
```

Patched for css:

```
\gdef\@thm#1#2{\refstepcounter{#1}%
\LWR@forcenewpage% lwarp
    \BlockClass{theorembody\LWR@thisthmstyle}% lwarp
   \trivlist
   \@topsep \theorempreskipamount               % used by first \item
   \@topsepadd \theorempostskipamount           % used by \@endparenv
   \@ifnextchar [%
   {\@ythm{#1}{#2}}%
   {\@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}}

\gdef\@endtheorem{%
\endtrivlist
\endBlockClass
}
```

File 280 `lwarp-threeparttable.sty`

§ 372 Package threeparttable

(*Emulates or patches code by* Donald Arseneau.)

Pkg `threeparttable` **threeparttable** is emulated.

Table note are contained inside a css `<div>` of class `tnotes`. If **enumitem** is used, the note item labels are also individually highlighted with an additional css `<span>` of class `tnoteitemheader`, otherwise they are plain text.

for HTML output:

```
\LWR@ProvidesPackageDrop{threeparttable}
```

`\LWR@printtablenote` {⟨*text*⟩}

Prints the table note item header inside a css class of `tnoteitemheader`.

```
\newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
```

Env `threeparttable` [⟨*alignment*⟩] To emulate threeparttable:

```
\newenvironment*{threeparttable}[1][b]{}{}
```

Env `tablenotes` [⟨*options*⟩]

```
\newenvironment*{tablenotes}[1][]
{%
\LWR@forcenewpage
\BlockClass{tnotes}%
\ltx@ifpackageloaded{enumitem}{%
\setlist[description]{format=\LWR@printtablenote}%
}{}%
\description%
}
{%
\enddescription%
\endBlockClass%
}
```

`\tnote` {⟨*text*⟩}

```
\newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

File 281 `lwarp-thumb.sty`

§ 373 Package thumb

Pkg `thumb` **thumb** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{thumb}
```

```
\newcommand*{\Overviewpage}{}
\newlength{\thumbheight}
\newlength{\thumbwidth}
```

File 282 `lwarp-thumbs.sty`

§ 374 Package thumbs

Pkg `thumbs` **thumbs** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{thumbs}
```

```
\newcommand{\addthumb}[4]{}
\newcommand{\addtitlethumb}[5]{}
\newcommand{\stopthumb}{}
\newcommand{\continuethumb}{}
```

```
\newcommand{\thumbsoverview}[1]{}
\newcommand{\thumbsoverviewback}[1]{}
\newcommand{\thumbsoverviewverso}[1]{}
\newcommand{\thumbsoverviewdouble}[1]{}
\newcommand{\thumbnewcolumn}{}
\newcommand{\addthumbsoverviewtocontents}[2]{}
\newcommand{\thumbsnophantom}{}
```

File 283 `lwarp-tikz.sty`

§ 375 Package

tikz

(*Emulates or patches code by* Till Tantau.)

Pkg `tikz` **tikz** is supported.

⚠ displaymath and matrices

If using display math with `tikzpicture` or `\tikz`, along with matrices with the `&` character, the document must be modified as follows:

```
\usepackage{tikz}
\tikzset{every picture/.style={ampersand replacement=\&}}
```

and each instance of `&` in the **tikz** expression must be replaced with `\&`.

Accept all options for **lwarp-tikz**:

```
\LWR@ProvidesPackagePass{tikz}[2015/08/07]
```

catcodes

lwarp changes the catcode of $ for its own use. The Ti*k*z `babel` library temporarily changes catcodes back to normal for Ti*k*z's use. **tikz** v3.0.0 introduced the `babel` library which handles catcode changes. For older versions, **lwarp** must change $'s catcode itself.

Also see:
https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded

```
\newbool{LWR@tikzbabel}

\@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
{\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
{\boolfalse{LWR@tikzbabel}}
```

Env `pgfpicture` The `\pgfpicture` environment is enclosed inside a `\lateximage`. Enclose the low-

level \pgfpicture in a lateximage. This is also used by the higher-level \tikz and tikzpicture.

```
\preto\pgfpicture{%
    \begin{lateximage}%
    \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
    {}%
    {\catcode`\$=3}% dollar sign is math shift
}

\appto\endpgfpicture{\end{lateximage}}
```

Ti*k*z is placed inside an SVG image, so use the original meanings of the following:

```
\LetLtxMacro\pgfutil@minipage\LWR@print@minipage
\let\pgfutil@endminipage\endLWR@print@minipage

\let\pgfutil@raggedleft\LWR@print@raggedleft
\let\pgfutil@raggedright\LWR@print@raggedright

\def\pgfutil@font@tiny{\LWR@printtiny}
\def\pgfutil@font@scriptsize{\LWR@printscriptsize}
\def\pgfutil@font@footnotesize{\LWR@printfootnotesize}
\def\pgfutil@font@small{\LWR@print@small}
\def\pgfutil@font@normalsize{\LWR@print@normalsize}
\def\pgfutil@font@large{\LWR@printlarge}
\def\pgfutil@font@Large{\LWR@printLarge}
\def\pgfutil@font@huge{\LWR@printhuge}
\def\pgfutil@font@Huge{\LWR@printHuge}

\def\pgfutil@font@itshape{\LWR@origitshape}
\def\pgfutil@font@bfseries{\LWR@origbfseries}

\def\pgfutil@font@normalfont{\LWR@orignormalfont}
```

File 284 `lwarp-titleps.sty`

§ 376 Package titleps

(*Emulates or patches code by* Javier Bezos.)

Pkg `titleps` **titleps** is loaded and used by **lwarp** during HTML output. All user options and macros are ignored and disabled.

Discard all options for **lwarp-titleps**:

for HTML output:

```
\LWR@ProvidesPackageDrop{titleps}
```

\pagestyle and \thispagestyle are already disabled in the **lwarp** code.

\newpagestyle {⟨*name*⟩} [⟨*style*⟩] {⟨*commands*⟩}

```
\NewDocumentCommand{\newpagestyle}{m o m}{}
```

\renewpagestyle {⟨*name*⟩} [⟨*style*⟩] {⟨*commands*⟩}

```
\NewDocumentCommand{\renewpagestyle}{m o m}{}
```

\sethead [⟨*el*⟩] [⟨*ec*⟩] [⟨*er*⟩] {⟨*ol*⟩} {⟨*oc*⟩} {⟨*or*⟩}

```
\NewDocumentCommand{\sethead}{o o o m m m}{}
```

\setfoot [⟨*el*⟩] [⟨*ec*⟩] [⟨*er*⟩] {⟨*ol*⟩} {⟨*oc*⟩} {⟨*or*⟩}

```
\NewDocumentCommand{\setfoot}{o o o m m m}{}
```

\settitlemarks * {⟨*names*⟩}

```
\NewDocumentCommand{\settitlemarks}{s m}{}
```

\headrule

```
\newcommand*{\headrule}{}
```

\footrule

```
\newcommand*{\footrule}{}
```

\setheadrule {⟨*length*⟩}

```
\newcommand*{\setheadrule}[1]{}
```

\setfootrule {⟨*length*⟩}

```
\newcommand*{\setfootrule}[1]{}
```

\makeheadrule

```
\newcommand*{\makeheadrule}{}
```

\makefootrule

```
\newcommand*{\makefootrule}{}
```

\setmarkboth {⟨*code*⟩}

```
\newcommand{\setmarkboth}[1]{}
```

\widenhead

```
\NewDocumentCommand{\widenhead}{s o o m m}{}
```

\bottitlemarks

```
\newcommand*{\bottitlemarks}{}
```

\toptitlemarks

```
\newcommand*{\toptitlemarks}{}
```

\firsttitlemarks

```
\newcommand*{\firsttitlemarks}{}
```

\nexttitlemarks

```
\newcommand*{\nexttoptitlemarks}{}
```

\outertitlemarks

```
\newcommand*{\outertitlemarks}{}
```

\innertitlemarks

```
\newcommand*{\innertitlemarks}{}
```

\newtitlemark * {⟨*name*⟩}

```
\NewDocumentCommand{\newtitlemark}{s m}{}
```

\pretitlemark * {⟨*section*⟩} {⟨*text*⟩}

```
\NewDocumentCommand{\pretitlemark}{s m m}{}
```

\ifsamemark {⟨*group*⟩} {⟨*command*⟩} {⟨*true*⟩} {⟨*false*⟩}

```
\newcommand{\ifsamemark}[4]{}
```

\setfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
\NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}
```

\setfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}
```

\nextfloathead * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}
```

\nextfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨*extra*⟩} [⟨*which*⟩]

```
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}
```

\newmarkset {⟨*markset*⟩}

```
28 \newcommand{\newmarkset}[1]{}
```

\newextramark * {⟨*markset*⟩} {⟨*macro-name*⟩}

```
29 \NewDocumentCommand{\newextramarkset}{s m m}{}
```

\botextramarks {⟨*markset*⟩}

```
30 \newcommand{\botextramarks}[1]{}
```

\topextramarks {⟨*markset*⟩}

```
31 \newcommand{\topextramarks}[1]{}
```

\firstextramarks {⟨*markset*⟩}

```
32 \newcommand{\firstextramarks}[1]{}
```

\nextextramarks {⟨*markset*⟩}

```
33 \newcommand{\nexttopextramarks}[1]{}
```

\outerextramarks {⟨*markset*⟩}

```
34 \newcommand{\outerextramarks}[1]{}
```

\innerextramarks {⟨*markset*⟩}

```
35 \newcommand{\innerextramarks}[1]{}
```

File 285 `lwarp-titleref.sty`

§ 377 Package titleref

Pkg `titleref` **titleref** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{titleref}

\LetLtxMacro\titleref\nameref

\providecounter{LWR@currenttitle}

\newcommand*{\currenttitle}{%
    \addtocounter{LWR@currenttitle}{1}%
    \label{currenttitle\arabic{LWR@currenttitle}}%
    \nameref{currenttitle\arabic{LWR@currenttitle}}%
}

\newcommand*{\theTitleReference}[2]{}
```

File 286 `lwarp-titlesec.sty`

§ 378 Package titlesec

(*Emulates or patches code by* Javier Bezos.)

Pkg `titlesec` **titlesec** is emulated. All user options and macros are ignored and disabled.

Discard all options for **lwarp-titlesec**:

for HTML output:

```
\LWR@ProvidesPackageDrop{titlesec}
```

`\titlelabel` {⟨*label-format*⟩}

```
\newcommand*{\titlelabel}[1]{}
```

`\titleformat*` {⟨*command*⟩} {⟨*format*⟩}

`\titleformat` {⟨*command*⟩} [⟨*shape*⟩] {⟨*format*⟩} {⟨*label*⟩} {⟨*sep*⟩} {⟨*begfore*⟩} [⟨*after*⟩]

```
\newcommand\titleformat{%
  \@ifstar{\ttl@format@s}%
```

```
          {\ttl@format@i}}
\newcommand{\ttl@format@s}[1]{}
\NewDocumentCommand{\ttl@format@i}{m o m m m m o}{}
```

\chaptertitlename

```
\@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
\newcommand\chaptertitlename{\@chapapp}
```

\titlespacing * {⟨*command*⟩} {⟨*left*⟩} {⟨*before*⟩} {⟨*after*⟩} [⟨*right*⟩]

```
\NewDocumentCommand{\titlespacing}{s m m m m o}{}
```

\filright

```
\newcommand*{\filright}{}
```

\filcenter

```
\newcommand*{\filcenter}{}
```

\filleft

```
\newcommand*{\filleft}{}
```

\fillast

```
\newcommand*{\fillast}{}
```

\filinner

```
\newcommand*{\filinner}{}
```

\filouter

```
\newcommand*{\filouter}{}
```

\wordsep

```
\newcommand\wordsep{\fontdimen\tw@\font \@plus
  \fontdimen\thr@@\font \@minus \fontdimen4\font}
```

\titleline * [⟨*align*⟩] {⟨*material*⟩}

```
\NewDocumentCommand{\titleline}{s o m}{}
```

`\titlerule` [⟨*height*⟩]

```
\providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
\newcommand*{\ttl@rule}[1][]{}
\newcommand*{\ttl@row}[2][]{}
```

`\iftitlemeasuring` {⟨*true*⟩} {⟨*false*⟩}

```
\newcommand{\iftitlemeasuring}[2]{#2}
```

`\assignpagestyle` {⟨*command*⟩} {⟨*pagestyle*⟩}

```
\newcommand{\assignpagestyle}[2]{#2}
```

`\titleclass` {⟨*name*⟩} [⟨*startlevel*⟩] {⟨*class*⟩} [⟨*cmd*⟩]

```
\NewDocumentCommand{\titleclass}{m o m o}
```

File 287 `lwarp-titletoc.sty`

§ 379 Package titletoc

(*Emulates or patches code by* Javier Bezos.)

Pkg `titletoc` **titletoc** is emulated. All user options and macros are ignored and disabled.

Discard all options for **lwarp-titletoc**:

for HTML output:

```
\LWR@ProvidesPackageDrop{titletoc}
```

`\dottedcontents` {⟨*section*⟩} [⟨*left*⟩] {⟨*above*⟩} {⟨*label*⟩} {⟨*leader*⟩}

```
\NewDocumentCommand{\dottedcontents}{m o m m m}{}
```

`\titlecontents` * {⟨*section*⟩} [⟨*left*⟩] {⟨*above*⟩} {⟨*numbered*⟩} {⟨*numberless*⟩} {⟨*filler*⟩} [⟨*below or begin*⟩] [⟨*separator*⟩] [⟨*end*⟩]

```
\newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
\NewDocumentCommand{\ttl@tcstar}{m o m m m m o o o}{}
\NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{}
```

`\contentsmargin` [⟨*correction*⟩] {⟨*right*⟩}

```
\newcommand{\contentsmargin}[2][]{}
```

`\thecontentslabel`

```
\newcommand*{\thecontentslabel}{thecontentslabel}
```

\thecontentspage

```
\newcommand*{\thecontentspage}{thecontentspage}
```

\contentslabel [⟨*format*⟩] {⟨*space*⟩}

```
\newcommand{\contentslabel}[2][]{\thecontentslabel}
```

\contentspage [⟨*format*⟩]

```
\newcommand{\contentspage}[1][]{\thecontentspage}
```

\contentspush {⟨*text*⟩}

```
\newcommand{\contentspush}[1]{}
```

\contentsuse {⟨*name*⟩} {⟨*text*⟩}

```
\newcommand{\contentsuse}[2]{}
```

\startcontents [⟨*name*⟩]

```
\newcommand*{\startcontents}[1][]{}
```

\stopcontents [⟨*name*⟩]

```
\newcommand*{\stopcontents}[1][]{}
```

\resumecontents [⟨*name*⟩]

```
\newcommand*{\resumecontents}[1][]{}
```

\printcontents [⟨*name*⟩] {⟨*prefix*⟩} {⟨*start*⟩} {⟨*code*⟩}

```
\newcommand{\printcontents}[4][]{}
```

\startlist [⟨*name*⟩] {⟨*list*⟩}

```
\newcommand{\startlist}[2][]{}
```

\stoplist [⟨*name*⟩] {⟨*list*⟩}

```
\newcommand{\stoplist}[2][]{}
```

\resumelist [⟨*name*⟩] {⟨*list*⟩}

```
\newcommand{\resumelist}[2][]{}
```

\printlist [⟨*name*⟩] {⟨*list*⟩} {⟨*prefix*⟩} {⟨*code*⟩}

```
\newcommand{\printlist}[4][]{}
```

File 288 `lwarp-titling.sty`

§ 380 Package

titling

(*Emulates or patches code by* Peter Wilson.)

Pkg titling

package support
⚠ load order

lwarp supports the native LaTeX titling commands, and also supports the packages **authblk** and **titling**. If both are used, **authblk** should be loaded before **titling**.

\published and \subtitle

If using the **titling** package, additional titlepage fields for \published and \subtitle may be added by using \AddSubtitlePublished in the preamble. See section 62.8.

The various **titling** footnote restyling commands have no effect.

Pass all options to **lwarp-titling**:

for HTML output:

```
\LWR@ProvidesPackagePass{titling}
```

\@bsmtitlempty Patch \@bsmtitlempty:

```
\let\LWR@orig@bsmtitlempty\@bsmtitlempty
\renewcommand*{\@bsmtitlempty}{%
\LWR@orig@bsmtitlempty%
}
```

\keepthetitle Patch \keepthetitle:

```
\let\LWR@origkeepthetitle\keepthetitle
\renewcommand*{\keepthetitle}{%
\LWR@orig@keepthetitle%
}
```

\killtitle Patch \killtitle:

```
\let\LWR@origkilltitle\killtitle
\renewcommand*{\killtitle}{%
\LWR@orig@killtitle%
```

```
}
```

Env `titlingpage`

```
\renewenvironment*{titlingpage}
{%
```

Start an HTML titlepage div:

```
\LWR@printpendingfootnotes
\begin{titlepage}
```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
\LWR@maketitlesetup
\let\maketitle\LWR@titlingmaketitle
}
{
```

At the end of the environment, end the HTML titlepage div:

```
\end{titlepage}
}
```

Patch the pre/post title/author/date to add HTML tags, then initilize:

```

\pretitle{}
\posttitle{}

\preauthor{}
\postauthor{}

\predate{}
\postdate{}
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
\renewcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
\def\@makefnmark{\textsuperscript{\@thefnmark}}
```

`\thefootnote` ⇒ `\nameuse`{arabic}{footnote}, or
`\thefootnote` ⇒ `\nameuse`{fnsymbol}{footnote}

Redefine the footnote text:

```
\long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
\makethanksmark~%
```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒ \@thefnmark ⇒ \itshape a (or similar)

Print the text:

```
##1%
}% \@makefntext
}
```

\thanksfootmark

```
\renewcommand{\thanksfootmark}{%
%   \hb@xt@\thanksmarkwidth{\hfil\normalfont%
        \thanksscript{%
            \thanksfootpre \tamark \thanksfootpost%
        }%
%   }%
}
```

\maketitle HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the **titling** package is adapted, simplified, and modified for HTML output.

```
\renewcommand*{\maketitle}{%
```

An HTML titlepage is used for all classes.

```
\begin{titlepage}
```

Select which kind of footnote marks to use:

```
\@bsmarkseries
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
```

Close the HTML titlepage div:

```
\end{titlepage}
```

Reset the footnote counter:

```
\@bscontmark
}
```

\@maketitle Typesets the title, etc. Patched for HTML.

```
\DeclareDocumentCommand{\@maketitle}{}{%
    \maketitlehooka
    {
        \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
        \@bspretitle \@title \@bsposttitle
        \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
    }
    \maketitlehookb
    {
        \begin{BlockClass}{author}
        \renewcommand{\and}{
            \end{BlockClass}
            \begin{BlockClass}{oneauthor}
        }
        \begin{BlockClass}{oneauthor}
        \@bspreauthor \@author  \@bspostauthor
        \end{BlockClass}
        \end{BlockClass}
    }
    \maketitlehookc
    {
        \begin{BlockClass}{titledate}
        \@bspredate \@date \@bspostdate
        \end{BlockClass}
    }
    \maketitlehookd
}
```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
\renewcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
\@thanks
```

Select which kind of footnote marks to use:

```
\@bsmarkseries
```

Set up special patches:

```
\LWR@maketitlesetup
```

Typeset the title, etc:

```
\@maketitle
```

Immediately generate any \thanks footnotes:

```
\@thanks
```

Reset the footnote counter:

```
\@bscontmark
}
```

\thanksmarkseries {⟨*series*⟩}

Sets the type of footnote marks used by \thanks, where type is 'arabic', 'roman', 'fnsymbol', etc.

```
\renewcommand{\thanksmarkseries}[1]{%
\def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
}
```

Set default titlepage thanks footnote marks. See section 62.7.

```
\@ifclassloaded{memoir}{
   \thanksmarkseries{arabic}
}{% not memoir
\if@titlepage
   \thanksmarkseries{arabic}
\else
```

```
    \thanksmarkseries{fnsymbol}
\fi
}% not memoir
```

File 289 `lwarp-tocbasic.sty`

§ 381 Package tocbasic

(*Emulates or patches code by* Markus Kohm.)

Pkg `tocbasic` **tocbasic** is patched for use by **lwarp**.

This package may be loaded standalone, but is also loaded automatically if **koma-script** classes are in use. `\DeclareDocumentCommand` is used to overwrite the **koma-script** definitions.

for HTML output:

```
\LWR@ProvidesPackagePass{tocbasic}

\DeclareDocumentCommand{\usetocbasicnumberline}{o}{}
\DeclareDocumentCommand{\DeclareTOCStyleEntry}{o m m}{}
\DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{}
\DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{}
\DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m m}{}
\DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m m}{}
\DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{}
\DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{}
\DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{}
```

File 290 `lwarp-tocbibind.sty`

§ 382 Package tocbibind

(*Emulates or patches code by* Peter Wilson.)

Pkg `tocbibind` **tocbibind** is patched for use by **lwarp**.

placement and TOC options An index may be placed inline with other HTML text, or on its own HTML page:

Pkg `makeidx` **Inline, with a manual TOC entry:**

A commonly-used method to introduce an index in a LaTeX document:

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex
```

Pkg makeidx **On its own HTML page, with a manual TOC entry:**

```
\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex
```

Pkg tocbibind **Inline, with an automatic TOC entry:**

The **tocbibind** package may be used to automatically place an entry in the TOC.

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex
```

Pkg tocbibind **On its own HTML page, with an automatic TOC entry:**

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex
```

Opt tocbibind numindex
numbered index section

Use the **tocbibind** numindex option to generate a numbered index. Without this option, the index heading has no number.

Other packages, such as **imakeidx**, may also have options for including the index in the Table of Contents.

for HTML output:

```
\let\simplechapterdelim\relax

\LWR@ProvidesPackagePass{tocbibind}


\renewenvironment{theindex}%
{%
    \if@bibchapter
       \if@donumindex
           \chapter{\indexname}
```

```
        \else
          \if@dotocind
            \chapter*{\indexname}
            \addcontentsline{toc}{chapter}{\indexname}
          \else
            \chapter*{\indexname}
          \fi
        \fi
     \else
        \if@donumindex
            \section{\indexname}
        \else
          \if@dotocind
            \section*{\indexname}
            \addcontentsline{toc}{\@tocextra}{\indexname}
          \else
            \section*{\indexname}
          \fi
        \fi
     \fi
\let\item\LWR@indexitem%
\let\subitem\LWR@indexsubitem%
\let\subsubitem\LWR@indexsubsubitem%
}{}
```

The following code is shared by **anonchap**.

```
\DeclareDocumentCommand{\simplechapter}{O{\@empty}}{%
    \def\@chapcntformat##1{%
        #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
    }%
}

\DeclareDocumentCommand{\restorechapter}{}{%
\let\@chapcntformat\@seccntformat%
}
```

File 291 `lwarp-tocenter.sty`

§ 383 Package

tocenter

Pkg `tocenter`

tocenter is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{tocenter}

\NewDocumentCommand{\ToCenter}{s o m m}{}
```

```
3 \NewDocumentCommand{\FromMargins}{s o m m m m}{}
```

File 292 `lwarp-tocloft.sty`

§ 384 Package

tocloft

(*Emulates or patches code by* Peter Wilson.)

Pkg tocloft

tocloft is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

Pkg tocloft

⚠ **tocloft** & other packages

If using **tocloft** with **tocbibind**, **anonchap**, **fncychap**, or other packages which change chapter title formatting, load **tocloft** with its `titles` option, which tells **tocloft** to use standard LaTeX commands to create the titles, allowing other packages to work with it.

Discard all options for **lwarp-tocloft**:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{tocloft}
```

`\tocloftpagestyle` {⟨*style*⟩}

```
2 \newcommand{\tocloftpagestyle}[1]{}
```

`\cftmarktoc`

```
3 \newcommand*{\cftmarktoc}{}
```

`\cfttoctitlefont`

```
4 \newcommand*{\cfttoctitlefont}{}
```

`\cftaftertoctitle`

```
5 \newcommand*{\cftaftertoctitle}{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}
```

`\cftmarklof`

```
8 \newcommand*{\cftmarklof}{}
```

`\cftloftitlefont`

```
\newcommand*{\cftloftitlefont}{}

\cftafterloftitle
\newcommand*{\cftafterloftitle}{}

\newlength{\cftbeforeloftitleskip}
\newlength{\cftafterloftitleskip}

\cftmarklot
\newcommand*{\cftmarklot}{}

\cftlottitlefont
\newcommand*{\cftlottitlefont}{}

\cftafterlottitle
\newcommand*{\cftafterlottitle}{}

\newlength{\cftbeforelottitleskip}
\newlength{\cftafterlottitleskip}

\cftdot
\providecommand*{\cftdot}{.}

\cftdotsep
\providecommand*{\cftdotsep}{1}

\cftnodots
\providecommand*{\cftnodots}{5000}

\cftdotfill {⟨sep⟩}
\providecommand{\cftdotfill}[1]{}

\cftsetpnumwidth {⟨length⟩}
\DeclareDocumentCommand{\cftsetpnumwidth}{m}{}

\cftsetrmarg {⟨length⟩}
```

```
\DeclareDocumentCommand{\cftsetrmarg}{m}{}
```

`\cftpnumalign` `{`⟨*alignment*⟩`}`

```
\DeclareDocumentCommand{\cftpnumalign}{m}{}

\LWR@providelength{\cftparskip}
```

The part-related items are also provided by **memoir**:

```
\LWR@providelength{\cftbeforepartskip}
\LWR@providelength{\cftpartindent}
\LWR@providelength{\cftpartnumwidth}
\providecommand*{\cftpartfont}{}
\providecommand*{\cftpartpresnum}{}
\providecommand*{\cftpartaftersnum}{}
\providecommand*{\cftpartaftersnumb}{}
\providecommand*{\cftpartleader}{}
\providecommand*{\cftpartdotsep}{1}
\providecommand*{\cftpartpagefont}{}
\providecommand*{\cftpartafterpnum}{}
```

memoir uses the full name “chapter” instead of “chap”:

```
\LWR@providelength{\cftbeforechapskip}
\LWR@providelength{\cftchapindent}
\LWR@providelength{\cftchapnumwidth}
\newcommand*{\cftchapfont}{}
\newcommand*{\cftchappresnum}{}
\newcommand*{\cftchapaftersnum}{}
\newcommand*{\cftchapaftersnumb}{}
\newcommand*{\cftchapleader}{}
\newcommand*{\cftchapdotsep}{1}
\newcommand*{\cftchappagefont}{}
\newcommand*{\cftchapafterpnum}{}
```

The following do not appear in **memoir**:

```
\LWR@providelength{\cftbeforesecskip}
\LWR@providelength{\cftsecindent}
\LWR@providelength{\cftsecnumwidth}
\newcommand*{\cftsecfont}{}
\newcommand*{\cftsecpresnum}{}
\newcommand*{\cftsecaftersnum}{}
\newcommand*{\cftsecaftersnumb}{}
\newcommand*{\cftsecleader}{}
\newcommand*{\cftsecdotsep}{1}
\newcommand*{\cftsecpagefont}{}
\newcommand*{\cftsecafterpnum}{}
```

```
\LWR@providelength{\cftbeforesubsecskip}
\LWR@providelength{\cftsubsecindent}
\LWR@providelength{\cftsubsecnumwidth}
\newcommand*{\cftsubsecfont}{}
\newcommand*{\cftsubsecpresnum}{}
\newcommand*{\cftsubsecaftersnum}{}
\newcommand*{\cftsubsecaftersnumb}{}
\newcommand*{\cftsubsecleader}{}
\newcommand*{\cftsubsecdotsep}{1}
\newcommand*{\cftsubsecpagefont}{}
\newcommand*{\cftsubsecafterpnum}{}

\LWR@providelength{\cftbeforesubsubsecskip}
\LWR@providelength{\cftsubsubsecindent}
\LWR@providelength{\cftsubsubsecnumwidth}
\newcommand*{\cftsubsubsecfont}{}
\newcommand*{\cftsubsubsecpresnum}{}
\newcommand*{\cftsubsubsecaftersnum}{}
\newcommand*{\cftsubsubsecaftersnumb}{}
\newcommand*{\cftsubsubsecleader}{}
\newcommand*{\cftsubsubsecdotsep}{1}
\newcommand*{\cftsubsubsecpagefont}{}
\newcommand*{\cftsubsubsecafterpnum}{}

\LWR@providelength{\cftbeforeparaskip}
\LWR@providelength{\cftparaindent}
\LWR@providelength{\cftparanumwidth}
\newcommand*{\cftparafont}{}
\newcommand*{\cftparapresnum}{}
\newcommand*{\cftparaaftersnum}{}
\newcommand*{\cftparaaftersnumb}{}
\newcommand*{\cftparaleader}{}
\newcommand*{\cftparadotsep}{1}
\newcommand*{\cftparapagefont}{}
\newcommand*{\cftparaafterpnum}{}

\LWR@providelength{\cftbeforesubparaskip}
\LWR@providelength{\cftsubparaindent}
\LWR@providelength{\cftsubparanumwidth}
\newcommand*{\cftsubparafont}{}
\newcommand*{\cftsubparapresnum}{}
\newcommand*{\cftsubparaaftersnum}{}
\newcommand*{\cftsubparaaftersnumb}{}
\newcommand*{\cftsubparaleader}{}
\newcommand*{\cftsubparadotsep}{1}
\newcommand*{\cftsubparapagefont}{}
\newcommand*{\cftsubparaafterpnum}{}

\LWR@providelength{\cftbeforefigskip}
```

```
\LWR@providelength{\cftfigindent}
\LWR@providelength{\cftfignumwidth}
\newcommand*{\cftfigfont}{}
\newcommand*{\cftfigpresnum}{}
\newcommand*{\cftfigaftersnum}{}
\newcommand*{\cftfigaftersnumb}{}
\newcommand*{\cftfigleader}{}
\newcommand*{\cftfigdotsep}{1}
\newcommand*{\cftfigpagefont}{}
\newcommand*{\cftfigafterpnum}{}

\LWR@providelength{\cftbeforesubfigskip}
\LWR@providelength{\cftsubfigindent}
\LWR@providelength{\cftsubfignumwidth}
\newcommand*{\cftsubfigfont}{}
\newcommand*{\cftsubfigpresnum}{}
\newcommand*{\cftsubfigaftersnum}{}
\newcommand*{\cftsubfigaftersnumb}{}
\newcommand*{\cftsubfigleader}{}
\newcommand*{\cftsubfigdotsep}{1}
\newcommand*{\cftsubfigpagefont}{}
\newcommand*{\cftsubfigafterpnum}{}

\LWR@providelength{\cftbeforetabskip}
\LWR@providelength{\cfttabindent}
\LWR@providelength{\cfttabnumwidth}
\newcommand*{\cfttabfont}{}
\newcommand*{\cfttabpresnum}{}
\newcommand*{\cfttabaftersnum}{}
\newcommand*{\cfttabaftersnumb}{}
\newcommand*{\cfttableader}{}
\newcommand*{\cfttabdotsep}{1}
\newcommand*{\cfttabpagefont}{}
\newcommand*{\cfttabafterpnum}{}

\LWR@providelength{\cftbeforesubtabskip}
\LWR@providelength{\cftsubtabindent}
\LWR@providelength{\cftsubtabnumwidth}
\newcommand*{\cftsubtabfont}{}
\newcommand*{\cftsubtabpresnum}{}
\newcommand*{\cftsubtabaftersnum}{}
\newcommand*{\cftsubtabaftersnumb}{}
\newcommand*{\cftsubtableader}{}
\newcommand*{\cftsubtabdotsep}{1}
\newcommand*{\cftsubtabpagefont}{}
\newcommand*{\cftsubtabafterpnum}{}

\DeclareDocumentCommand{\cftsetindents}{m m m}{}
```

```
\newcommand{\pagenumbersoff}[1]{}
\newcommand{\pagenumberson}[1]{}
```

`\newlistentry` [⟨*within*⟩] {⟨*counter*⟩} {⟨*ext*⟩} {⟨*level-1*⟩}

```
\DeclareDocumentCommand{\newlistentry}{o m m m}
{%
\LWR@traceinfo{newlistentry #2 #3 #4}%
\IfValueTF{#1}%
{%
    \@ifundefined{c@#2}{%
        \newcounter{#2}[#1]%
        \expandafter\edef\csname the#2\endcsname{%
            \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}%
        }%
    }{}%
}%
{%
    \@ifundefined{c@#2}{%
        \newcounter{#2}%
    }{}%
}%
\@namedef{l@#2}##1##2{%
    \hypertocfloat{1}{#2}{#3}{##1}{##2}%
    \def\cftwhatismyname{#2}% from memoir
}%
\expandafter\newlength\csname cftbefore#2skip\endcsname%
\expandafter\newlength\csname cft#2indent\endcsname%
\expandafter\newlength\csname cft#2numwidth\endcsname%
\@namedef{cft#2font}{}%
\@namedef{cft#2presnum}{}%
\@namedef{cft#2aftersnum}{}%
\@namedef{cft#2aftersnumb}{}%
\@namedef{cft#2leader}{}%
\@namedef{cft#2dotsep}{1}%
\@namedef{cft#2pagefont}{}%
\@namedef{cft#2afterpnum}{}%
\@namedef{toclevel@#2}{#4}%
\@namedef{cft#2fillnum}##1{}%
\LWR@traceinfo{newlistentry done}%
}
```

`\newlistof` [⟨*within*⟩] {⟨*type*⟩} {⟨*ext*⟩} {⟨*listofname*⟩}

Emulated through the `\newfloat` mechanism.

```
\DeclareDocumentCommand{\newlistof}{o m m m}
{%
\IfValueTF{#1}
```

```
{\newlistentry[#1]{#2}{#3}{0}}
{\newlistentry{#2}{#3}{0}}
\@namedef{ext@#2}{#3}
\@ifundefined{c@#3depth}{\newcounter{#3depth}}{}
\setcounter{#3depth}{1}
\@namedef{cftmark#3}{}
\@namedef{listof#2}{\listof{#2}{#4}}
\@namedef{@cftmake#3title}{}
\expandafter\newlength\csname cftbefore#3titleskip\endcsname
\expandafter\newlength\csname cftafter#3titleskip\endcsname
\@namedef{cft#3titlefont}{}
\@namedef{cftafter#3title}{}
\@namedef{cft#3prehook}{}
\@namedef{cft#3posthook}{}
}
```

`\cftchapterprecis` {⟨*text*⟩}

```
\newcommand{\cftchapterprecis}[1]{%
  \cftchapterprecishere{#1}
  \cftchapterprecistoc{#1}}
\newcommand{\cftchapterprecishere}[1]{%
  \begin{quote}\textit{#1}\end{quote}}
\newcommand{\cftchapterprecistoc}[1]{
  \addtocontents{toc}{%
  {
     \protect\begin{quote}#1\protect\end{quote}}
  }
}
```

File 293 `lwarp-tocstyle.sty`

§ 385 Package

tocstyle

Pkg `tocstyle` **tocstyle** is ignored.

⚠ Not fully tested! Please send bug reports!

for HTML output:

```
\LWR@ProvidesPackageDrop{tocstyle}
```

```
\newcommand*{\usetocstyle}[2][]{}
\newcommand*{\deactivatetocstyle}[1][]{}
\newcommand*{\reactivatetocstyle}[1][]{}
\NewDocumentCommand{\settocfeature}{o o m m}{}
\NewDocumentCommand{\settocstylefeature}{o m m}{}
\NewDocumentCommand{\newtocstyle}{o o m m}{}
\newcommand*{\aliastoc}[2]{}
```

```
\newcommand*{\showtoc}[2][]{}
\newcommand{\iftochasdepth}[4]{}
```

File 294 `lwarp-todo.sty`

§ 386 Package

todo

(*Emulates or patches code by* Federico Garcia.)

Pkg `todo` **todo** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{todo}

\renewcommand\todoitem[2]{%
    \refstepcounter{todo}%
    \item[%
        \HTMLunicode{2610} \quad
        \ref{todopage:\thetodo}
     ] : {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
    \label{todolbl:\thetodo}%
}%

\renewcommand\doneitem[2]{%
    \stepcounter{todo}%
    \item[%
        \HTMLunicode{2611} \quad
        \ref{todopage:\thetodo}
    ] \@nameuse{@done\the\c@todo}:
        {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
}

\xpatchcmd{\@displaytodo}
    {\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch @displaytodo.}}

\xpatchcmd{\@displayfulltodo}
    {\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch @displayfulltodo.}}

\patchcmd{\todoenv}{\itshape see text.}{\textit{see text.}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch todoenv.}}

\patchcmd{\astodos}{\todoformat #1}{\todoformat \textbf{#1}}{}
    {\PackageWarning{lwarp-todo}{Unable to patch astodos.}}

\AtBeginDocument{
\crefname{todo}{todo}{todos}
```

```
\Crefname{todo}{Todo}{Todos}
}
```

File 295 `lwarp-todonotes.sty`

§ 387 Package

todonotes

(*Emulates or patches code by* Henrik Skov Midtiby.)

Pkg `todonotes`

todonotes is emulated.

The documentation for **todonotes** and **luatodonotes** have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

for HTML output:

```
\LWR@ProvidesPackagePass{todonotes}

\if@todonotes@disabled
\else

\newcommand{\ext@todo}{tdo}

\renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}

\let\LWRTODONOTES@orig@todototoc\todototoc

\renewcommand*{\todototoc}{%
\phantomsection%
\LWRTODONOTES@orig@todototoc%
}

\renewcommand{\@todonotes@drawMarginNoteWithLine}{
\fcolorbox
    {\@todonotes@currentbordercolor}
    {\@todonotes@currentbackgroundcolor}
    {\arabic{@todonotes@numberoftodonotes}}
\marginpar{\@todonotes@drawMarginNote}
}

\renewcommand{\@todonotes@drawInlineNote}{%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \if@todonotes@authorgiven%
        {\@todonotes@author:\,}%
```

```
        \fi%
        \@todonotes@text%
    }%
}

\renewcommand{\@todonotes@drawMarginNote}{%
    \if@todonotes@authorgiven%
        \@todonotes@author\par%
    \fi%
    \arabic{@todonotes@numberoftodonotes}: %
    \fcolorbox%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentbackgroundcolor}%
    {%
        \@todonotes@sizecommand%
        \@todonotes@text %
    }%
}%

\renewcommand{\@todonotes@drawLineToRightMargin}{}

\renewcommand{\@todonotes@drawLineToLeftMargin}{}

\renewcommand{\missingfigure}[2][]{%
\setkeys{todonotes}{#1}%
\addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
\fcolorboxBlock%
    {\@todonotes@currentbordercolor}%
    {\@todonotes@currentfigcolor}%
    {%
        \setlength{\fboxrule}{4pt}%
        \fcolorbox{red}{white}{Missing figure} \quad #2%
    }
}

\LetLtxMacro\LWRTODONOTES@orig@todo\@todo

\RenewDocumentCommand{\@todo}{o m}{%
\begingroup%
\renewcommand*{\phantomsection}{}%
\IfValueTF{#1}{%
    \LWRTODONOTES@orig@todo[#1]{#2}%
}{%
    \LWRTODONOTES@orig@todo{#2}%
}
\endgroup%
}

\fi% \if@todonotes@disabled
```

File 296 `lwarp-transparent.sty`

§ 388 Package transparent

(*Emulates or patches code by* Heiko Oberdiek.)

Pkg `transparent` Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

⚠ Not XeLaTeX! Note that **transparent** does not work with XeLaTeX.

for HTML output: Discard all options for **lwarp-transparent**:

```
\LWR@ProvidesPackageDrop{transparent}

\newcommand*{\transparent}[1]{\edef\LWR@opacity{#1}}

\newcommand*{\texttransparent}[2]{%
\begingroup%
\transparent{#1}%
\InlineClass[opacity: #1]{transparent}{#2}%
\endgroup%
}
```

File 297 `lwarp-trimclip.sty`

§ 389 Package trimclip

Pkg `trimclip` **trimclip** is nullified.

for HTML output:

```
\LWR@ProvidesPackageDrop{trimclip}
```

The third argument, the text, is not touched. This allows `\bgroup` / `\egroup`, and verbatim content.

```
\csdef{trimbox}{\@ifstar\@gobble\@gobble}
\csletcs{trimbox*}{trimbox}
\def\endtrimbox{}
\csletcs{endtrimbox*}{endtrimbox}

\csletcs{clipbox}{trimbox}
\csletcs{clipbox*}{trimbox}
```

```
\csletcs{endclipbox}{endtrimbox}
\csletcs{endclipbox*}{endtrimbox}

\csletcs{marginbox}{trimbox}
\csletcs{marginbox*}{trimbox}
\csletcs{endmarginbox}{endtrimbox}
\csletcs{endmarginbox*}{endtrimbox}
```

File 298 `lwarp-trivfloat.sty`

§ 390 Package

trivfloat

(*Emulates or patches code by* Joseph Wright.)

Pkg `trivfloat` **trivfloat** is forced to use the built-in **lwarp** emulation for floats.

Discard all options for **lwarp-trivfloat**. This tells **trivfloat** not to use **floatrow** or **memoir**.

To create a new float type and change its name:

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

```
\LWR@ProvidesPackageDrop{trivfloat}
\LWR@origRequirePackage{trivfloat}
```

`\tfl@chapter@fix` Nullified at the beginning of the document. Is used by **trivfloat** to correct float chapter numbers, but is not needed for **lwarp**.

for HTML output:

```
\begin{warpHTML}
\AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}
\end{warpHTML}
```

§ 390.1 Combining \newfloat, \trivfloat, and algorithmicx

for HTML & PRINT:

```
6 \begin{warpall}
```

For both print and HTML output:

⚠ When using **float**, **trivfloat**, or **algorithmicx** at the same time, be aware of conflicting file usage. **algorithmicx** uses `.loa`. **trivfloat** by default starts with `.loa` and goes up for additional floats, skipping `.lof` and `.lot`.

⚠ When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid `.loa` used by **algorithmicx**, and any files used by **trivfloat**. Also avoid using `.lof` and `.lot`.

⚠ When using `\trivfloat`, you may force it to avoid conflicting with **algorithmicx** by starting **trivfloat**'s file extensions with `.lob`:

```
\makeatletter
\setcounter{tfl@float@cnt}{1} % start trivfloats with .lob
\makeatletter
```

```
7 \end{warpall}
```

File 299 `lwarp-turnthepage.sty`

§ 391 Package turnthepage

Pkg `turnthepage` **turnthepage** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{turnthepage}
```

```
2 \newcommand{\turnthepage}{}
```

File 300 `lwarp-twoup-gen.sty`

§ 392 Package twoup-gen

Pkg `twoup-gen` **twoup-gen** is ignored.

This package is used by MikTeX for **lwarp-2up**.

for HTML output:

```
\LWR@ProvidesPackageDrop{twoup-gen}

\LWR@origRequirePackage{lwarp-2up}
```

File 301 `lwarp-typearea.sty`

§ 393 Package typearea

(*Emulates or patches code by* Markus Kohm.)

Pkg `typearea` **typearea** is emulated.

This package may be loaded standalone, but is also loaded automatically if **koma-script** classes are in use. `\DeclareDocumentCommand` is used to overwrite the **koma-script** definitions.

for HTML output:

```
\LWR@ProvidesPackageDrop{typearea}

\DeclareDocumentCommand{\typearea}{o m}{}
\DeclareDocumentCommand{\recalctypearea}{}{}
\@ifundefined{footheight}{\newlength\footheight}{}
\DeclareDocumentCommand{\areaset}{o m m}{}
\DeclareDocumentCommand{\activateareas}{}{}
\DeclareDocumentCommand{\storeareas}{m}{}
\DeclareDocumentCommand{\BeforeRestoreareas}{s m}{}
\DeclareDocumentCommand{\AfterRestoreareas}{s m}{}
\DeclareDocumentCommand{\AfterCalculatingTypearea}{s m}{}
\DeclareDocumentCommand{\AfterSettingArea}{s m}{}
```

File 302 `lwarp-ulem.sty`

§ 394 Package ulem

(*Emulates or patches code by* Donald Arseneau.)

Pkg `ulem` Emulated.

for HTML output: Emulate the original package:

```
\ProvidesPackage{lwarp-ulem}
```

Original **lwarp** definitions:

```
\LetLtxMacro\LWR@ulemorigemph\emph
```

```
\LetLtxMacro\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using css:

```
\NewDocumentCommand{\uline}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:underline; text-decoration-skip: auto}%
    {uline}{#1}%
}

\NewDocumentCommand{\uuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline; text-decoration-skip: auto;%
        text-decoration-style:double%
    }%
    {uuline}{#1}%
}

\NewDocumentCommand{\uwave}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline; text-decoration-skip: auto;%
        text-decoration-style:wavy%
    }%
    {uwave}{#1}%
}

\NewDocumentCommand{\sout}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:line-through}%
    {sout}{#1}%
}

\NewDocumentCommand{\xout}{+m}{%
\LWR@HTMLtextstyle%
    {text-decoration:line-through}%
    {xout}{#1}%
}

\NewDocumentCommand{\dashuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;%
        text-decoration-skip: auto;%
        text-decoration-style:dashed%
    }%
    {dashuline}{#1}%
}
```

```
\NewDocumentCommand{\dotuline}{+m}{%
\LWR@HTMLtextstyle%
    {%
        text-decoration:underline;%
        text-decoration-skip: auto;%
        text-decoration-style: dotted%
    }%
    {dotuline}{#1}%
}
```

Nullified parameters:

```
\NewDocumentCommand{\ULthickness}{}{}
\newlength{\ULdepth}
```

Nullified/emulated macros:

```
\NewDocumentCommand{\markoverwith}{m}{}
\NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

`\useunder` only works with `\textbf`, etc, but not `\bfseries`, etc.

```
\NewDocumentCommand{\useunder}{m m m}{%
\relax%
\ifx\relax#3\relax\else % argumentative command
    \def#3{#1}\MakeRobust{#3}\fi
}
```

Triggered by package options, also available for the users:

```
\newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
\newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
\ULforem% default
```

Package options:

```
\DeclareOption{normalem}{\normalem}
\DeclareOption{ULforem}{\ULforem}
\DeclareOption{normalbf}{}
\DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}

\DeclareOption*{}
\ProcessOptions\relax% original LaTeX code
```

File 303 `lwarp-underscore.sty`

§ 395 Package underscore

Pkg `underscore` **underscore** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{underscore}
```

File 304 `lwarp-units.sty`

§ 396 Package units

(*Emulates or patches code by* Axel Reichert.)

Pkg `units` **units** is patched for use by **lwarp**.

for HTML output:

```
1 \LWR@ProvidesPackagePass{units}[1998/08/04]
```

For Mathjax:

```
2 \CustomizeMathJax{\newcommand{\unit}[2][]{#1 #2}}
3 \CustomizeMathJax{\newcommand{\unitfrac}[3][]{#1 #2/#3}}
```

File 305 `lwarp-upref.sty`

§ 397 Package upref

Pkg `upref` Ignored.

for HTML output: Discard all options for **lwarp-upref**:

```
1 \LWR@ProvidesPackageDrop{upref}
```

File 306 `lwarp-url.sty`

§ 398 Package url

(*Emulates or patches code by* Donald Arseneau.)

Pkg url **url** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{url}
```

url uses math mode to print its string inside a group, so the original meaning of math is restored first.

```
\LetLtxMacro\LWR@url@origUrl@FormatString\Url@FormatString

\renewcommand*{\Url@FormatString}{%
    \InlineClass{verbatim}{%
        \LWR@restoreorigformatting%
        \LWR@url@origUrl@FormatString%
    }%
}
```

File 307 `lwarp-verse.sty`

§ 399 Package verse

(*Emulates or patches code by* Peter Wilson.)

Pkg verse **verse** is supported and patched by **lwarp**.

for HTML output: Pass all options for **lwarp-verse**:

```
\LWR@ProvidesPackagePass{verse}
```

When using **verse** or **memoir**, always place a \\ after each line.

\attrib The documentation for the **verse** and **memoir** packages suggest defining an \attrib command, which may already exist in current documents, but it will only work for print output. **lwarp** provides \attribution, which works for both print and HTML output. To combine the two so that \attrib is used for print and \attribution is used for HTML:

```
\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}
```

Len \vleftskip
Len \vleftmargini
Len \HTMLvleftskip
Len \HTMLleftmargini

These lengths are used by **verse** and **memoir** to control the left margin, and they may already be set by the user for print output. New lengths \HTMLvleftskip and \HTMLleftmargini are provided to control the margins in HTML output. These new

lengths may be set by the user before any `verse` environment, and persist until they are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

⚠ spacing Horizontal spacing relies on **pdftotext**’s ability to discern the layout (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

Env `verse` The `verse` environment will be placed inside a HTML `<pre>`.

```
\AfterEndPreamble{
\LWR@traceinfo{Patching verse.}
```

At the beginning of the `verse` environment:

```
\AtBeginEnvironment{verse}
{%
```

Use the original `list` environment inside a `<pre>` to attempt to preserve formatting.

```
\LWR@restoreoriglists%
```

Pkg `verse` Cls `memoir` `\flagverse` Len `\vleftskip` The **verse** or **memoir** packages can place stanza numbers to the left with their `\flagverse` command. Do not allow them to go into the left margin, which would cause **pdfcrop** to crop the entire page further to the left:

```
\ifdef{\vleftskip}{%
\setlength{\vleftskip}{\HTMLvleftskip}
\setlength{\leftmargini}{\HTMLleftmargini}
}{}
\LWR@forcenewpage
\LWR@atbeginverbatim{3}{verse}%
}
```

After the end of the `verse` environment, which places the `<pre>` tag at the regular left margin:

```
\AtEndEnvironment{verse}{%
\leavevmode%
\LWR@afterendverbatim{1}%
}
```

Patch to place poemtitle inside an HTML `<span>` of class `poemtitle`:

```
\ifdef{\poemtitle}{
\DeclareDocumentCommand{\@vstypeptitle}{m}{%
   \vspace{\beforepoemtitleskip}%
   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
   \vspace{\afterpoemtitleskip}%
 }
}{}

\LWR@traceinfo{Finished patching verse.}
}% AfterEndPreamble
```

File 308 `lwarp-vertbars.sty`

§ 400 Package vertbars

(*Emulates or patches code by* Peter Wilson.)

Pkg `vertbars` **vertbars** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{vertbars}

\newlength{\barwidth}
\setlength{\barwidth}{0.4pt}
\newlength{\barspace}
\setlength{\barspace}{1em}

\newenvironment{vertbar}{
    \LWR@forcenewpage
    \LWR@forceminwidth{\barwidth}
    \begin{BlockClass}[%
        border-left: \LWR@printlength{\LWR@atleastonept} solid black ; %
        padding-left: \LWR@printlength{\barspace}%
    ]{vertbar}
}{
    \end{BlockClass}
}
```

File 309 `lwarp-vmargin.sty`

§ 401 Package vmargin

Pkg `vmargin` **vmargin** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{vmargin}
```

```
\newcommand*{\LWRVM@customsize}[2]{}
\newcommand*{\setpapersize}[2][]{\ifstrequal{#2}{custom}{\LWRVM@customsize}{}}
\newcommand*{\setmargins}[8]{}
\newcommand*{\setmarginsrb}[8]{}
\newcommand*{\setmargnohf}[4]{}
\newcommand*{\setmargnohfrb}[4]{}
\newcommand*{\setmarg}[4]{}
\newcommand*{\setmargrb}[4]{}
\newlength{\PaperWidth}
\setlength{\PaperWidth}{8.5in}
\newlength{\PaperHeight}
\setlength{\PaperHeight}{11in}
\newif\ifLandscape
```

File 310 `lwarp-vowel.sty`

§ 402 Package vowel

(*Emulates or patches code by* FUKUI REI.)

Pkg `vowel` **vowel** is patched for use by **lwarp**.

This package has been tested with **pdflatex** and the Type 1 TIPA fonts using the following package load sequence:

```
\usepackage[T3,T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[noenc]{tipa}
\usepackage{vowel}
```

for HTML output:

```
\LWR@ProvidesPackagePass{vowel}

\renewenvironment{vowel}[1][]
    {%
        \begin{lateximage}[(-vowel-~\packagediagramname)]%
        \@vowel[#1]%
    }
    {%
        \@@vowel%
        \end{lateximage}%
    }
```

File 311 `lwarp-vwcol.sty`

§ 403 Package

vwcol

(*Emulates or patches code by* Will Robertson.)

Pkg `vwcol` **vwcol** is patched for use with **lwarp**.

The `width` option is ignored. All `vwcol` environments adjust to 1–3 equal-width columns, depening on the width of the browser window.

The remaining options are supported, except for `lines` and `maxrecursion`.

for HTML output:

```
\LWR@ProvidesPackagePass{vwcol}
```

Factored from `\vwcol`. Each is given a style tag to append to the final style.

`\LWR@vwcol@addrule` {⟨*style tag*⟩}

```
\newcommand*{\LWR@vwcol@addrule}[1]{%
    \appto{\LWR@vwcolstyle}{%
        #1: %
        \LWR@printlength{\vwcol@rule} solid \LWR@origpound\LWR@vwcol@rulecolor ; %
    }%
}
```

`\LWR@vwcol@addrule` {⟨*style tag*⟩}

```
\newcommand*{\LWR@vwcol@addgap}[1]{%
    \appto{\LWR@vwcolstyle}{%
        #1: %
        \LWR@printlength{\vwcol@sep} ; %
    }%
}
```

Env `vwcol` {⟨*key/values*⟩}

Redefine the environment to add a HTML style. The style is built depending on the required options.

```
\renewenvironment*{vwcol}[1][]{%
```

New paragraph, and process the options:

```
\par\noindent%
\vwcolsetup{#1}%
```

Begin with no style:

```
\newcommand*{\LWR@vwcolstyle}{}
```

presep and postsep are created with HTML margins:

```
\if@vwcol@presep
    \appto{\LWR@vwcolstyle}{margin-left: 1em ; padding-left: .5em ; }
\fi
\if@vwcol@postsep
    \appto{\LWR@vwcolstyle}{margin-right: 1em ; padding-right: .5em ; }
\fi
```

sep becomes column-gap:

```
\ifdimgreater{\vwcol@sep}{1sp}{
    \LWR@vwcol@addgap{column-gap}
    \LWR@vwcol@addgap{-moz-column-gap}
    \LWR@vwcol@addgap{-webkit-column-gap}
}{}
```

rule become column-rule, while prerule and postrule become HTML borders:

```
\convertcolorspec{named}{\vwcol@rulecol}{HTML}\LWR@vwcol@rulecolor%
\ifdimgreater{\vwcol@rule}{0pt}{
    \ifdimless{\vwcol@rule}{1pt}{
        \setlength{\vwcol@rule}{1pt}
    }{}
    \LWR@vwcol@addrule{column-rule}
    \LWR@vwcol@addrule{-moz-column-rule}
    \LWR@vwcol@addrule{-webkit-column-rule}
    \if@vwcol@prerule\LWR@vwcol@addrule{border-left}\fi
    \if@vwcol@postrule\LWR@vwcol@addrule{border-right}\fi
}{}
```

Each of the justify options becomes a text-align. Indentation is added where appropriate.

```
\ifdefequal{\vwcol@justify}{\RaggedRight}{
    \appto{\LWR@vwcolstyle}{text-align: left ; }
    \ifdimgreater{\vwcol@parindent}{0pt}{
        \appto{\LWR@vwcolstyle}{%
            text-indent: \LWR@printlength{\vwcol@parindent} ; %
        }
    }{}
}{}
```

```
\ifdefequal{\vwcol@justify}{\RaggedLeft}{
    \appto{\LWR@vwcolstyle}{text-align: right ; }
}{}
```

```
\ifdefequal{\vwcol@justify}{\Centering}{
    \appto{\LWR@vwcolstyle}{text-align: center ; }
}{}
```

```
\ifdefequal{\vwcol@justify}{\justifying}{
    \appto{\LWR@vwcolstyle}{text-align: justify ; }
    \ifdimgreater{\vwcol@parindent}{0pt}{
        \appto{\LWR@vwcolstyle}{%
            text-indent: \LWR@printlength{\vwcol@parindent} ; %
        }
    }{}
}{}
```

Create the with the assembled style:

```
\BlockClass[\LWR@vwcolstyle]{multicols}
}
```

When the environment ends:

```
{
\endBlockClass
}
```

File 312 `lwarp-wallpaper.sty`

§ 404 Package wallpaper

(*Emulates or patches code by* Michael H.F. Wilkinson.)

Pkg `wallpaper` **wallpaper** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{wallpaper}

\newcommand*{\CenterWallPaper}[2]{}
\newcommand*{\ThisCenterWallPaper}[2]{}
\newcommand*{\TileWallPaper}[3]{}
\newcommand*{\ThisTileWallPaper}[3]{}
\newcommand*{\TileSquareWallPaper}[2]{}
\newcommand*{\ThisTileSquareWallPaper}[2]{}
\newcommand*{\ULCornerWallPaper}[2]{}
\newcommand*{\ThisULCornerWallPaper}[2]{}
\newcommand*{\LLCornerWallPaper}[2]{}
\newcommand*{\ThisLLCornerWallPaper}[2]{}
\newcommand*{\URCornerWallPaper}[2]{}
\newcommand*{\ThisURCornerWallPaper}[2]{}
\newcommand*{\LRCornerWallPaper}[2]{}
\newcommand*{\ThisLRCornerWallPaper}[2]{}
\newcommand*{\ClearWallPaper}{}
\newlength{\wpXoffset}
\newlength{\wpYoffset}
```

File 313 `lwarp-wasysym.sty`

§ 405 Package wasysym

Pkg `wasysym` **wasysym** does not work with **pdftotext**.

for HTML output:

```
\LWR@loadnever{wasysym}{textcomp, amssymb, amsfonts, mnsymbol, fdsymbol}
```

File 314 `lwarp-watermark.sty`

§ 406 Package watermark

(*Emulates or patches code by* Alexander I. Rozhenko.)

Pkg `watermark` **watermark** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{watermark}

\newcommand{\watermark}[1]{}
\newcommand{\leftwatermark}[1]{}
\newcommand{\rightwatermark}[1]{}
\newcommand{\thiswatermark}[1]{}
\newcommand{\thispageheading}[1]{}
```

File 315 `lwarp-widows-and-orphans.sty`

§ 407 Package widows-and-orphans

Pkg `widows-and-orphans` **widows-and-orphans** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{widows-and-orphans}[2018/09/01]

\NewDocumentCommand\WaOsetup{m}{}
\NewDocumentCommand\WaOparameters{}{}
\NewDocumentCommand\WaOignorenext{}{}
```

File 316 `lwarp-wrapfig.sty`

§ 408 Package wrapfig

(*Emulates or patches code by* Donald Arseneau.)

Pkg `wrapfig` **wrapfig** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{wrapfig}


\newcommand*{\LWR@wrapposition}{}

\newcommand*{\LWR@subwrapfigure}[2]{%
\renewcommand*{\LWR@wrapposition}{}%
\ifthenelse{%
    \equal{#1}{r}\OR\equal{#1}{R}\OR%
    \equal{#1}{o}\OR\equal{#1}{O}%
}%
{\renewcommand*{\LWR@wrapposition}{float:right}}%
{\renewcommand*{\LWR@wrapposition}{float:left}}%
\setlength{\LWR@templengthone}{#2}%
\LWR@BlockClassWP{%
    width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition; %
    margin:10pt%
}%
{%
    width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition%
}%
{marginblock}%
}


\NewDocumentEnvironment{wrapfigure}{o m o m}
{%
\LWR@subwrapfigure{#2}{#4}%
\captionsetup{type=figure}%
}
{%
\endLWR@BlockClassWP%
}


\NewDocumentEnvironment{wraptable}{o m o m}
{%
\LWR@subwrapfigure{#2}{#4}%
```

```
\captionsetup{type=table}%
}
{%
\endLWR@BlockClassWP%
}


\NewDocumentEnvironment{wrapfloat}{m o m o m}
{%
\LWR@subwrapfigure{#3}{#5}%
\captionsetup{type=#1}%
}
{%
\endLWR@BlockClassWP%
}

\newlength{\wrapoverhang}
```

File 317 `lwarp-xcolor.sty`

§ 409 Package

xcolor

(*Emulates or patches code by* Dr. Uwe Kern.)

Pkg `xcolor` **xcolor** is supported by lwarp.

§ 409.1 Limitations

`\colorboxBlock` and `\fcolorboxBlock` — `\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a `<span>`. These `<div>`s are set to `display: inline-block` so adjacent `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by **lwarp**'s core if **xcolor** is loaded.

background: `none` — `\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case only the frame is drawn, which can be useful for HTML.

color support — Color definitions, models, and mixing are fully supported without any changes required.

colored tables — `\rowcolors` is supported, except that the optional argument is ignored so far.

colored text and boxes — `\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

\color and \pagecolor

\color and \pagecolor are ignored. Use CSS or \textcolor where possible.

§ 409.2 Xcolor definitions: location and timing

The **lwarp** core and its **lwarp-xcolor** package are tightly integrated to allow comparable results for print, HTML and print inside an HTML lateximage. This requires a number of definitions and redefintions depending on whether each of **xcolor** and lateximage is being used, and whether print or HTML is being generated. Some of these actions are one-time when **xcolor** is loaded, and others are temporary as lateximage is used.

When xcolor is loaded in print mode: No special actions are taken at the time that **xcolor** is loaded in print mode, but see \AtBeginDocument below.

When lwarp-xcolor is loaded in HTML mode: **xcolor**'s original definitions are saved for later restoration. \LWR@restoreorigformatting is appended to restore these definitions for use inside a lateximage. New HTML-mode definitions are created for \textcolor, \pagecolor, \nopagecolor, \colorbox, \colorboxBlock, \fcolorbox, \fcolorboxBlock, and fcolorminipage.

\AtBeginDocument in print or HTML mode: See Section 81. If **xcolor** has been loaded, the print-mode \fcolorbox is modified to accept a background color of none, and additional definitions are created for **lwarp**'s new macros print-mode macros \colorboxBlock, \fcolorboxBlock, and fcolorminipage. The HTML versions of these macros will already have been created by **lwarp-xcolor** if it has been loaded.

For use inside an HTML lateximage, \LWR@restoreorigformatting is appended to temporarily set these functions to their print-mode versions.

In a lateximage in HTML mode: \LWR@restoreorigformatting temporarily restores the print-mode definitions of **xcolor**'s functions. See \LWR@restoreorigformatting on page 487.

\color:

Print: Used as-is.

HTML: Ignored by **pdftotext**, and will not appear.

HTML lateximage: Colors will appear in a lateximage.

\textcolor:

Print: Used as-is.

HTML: Redefined by **lwarp-xcolor**, page 927.

HTML lateximage: Remembers and reuses the print version.

\pagecolor:

Print: Used as-is.

HTML: Ignored.

HTML `lateximage`: Colors will be picked up in a `lateximage`.

`\nopagecolor`**:**

Print: Used as-is.

HTML: Ignored.

HTML `lateximage`: Colors will be picked up in a `lateximage`.

`\colorbox`**:**

Print: Used as-is.

HTML: Redefined by **lwarp-xcolor**, page 927.

HTML `lateximage`: Remembers and reuses the print version.

`\colorboxBlock`**:**

Print: Becomes `\colorbox`.

HTML: Newly defined by **lwarp-xcolor** to use a `<div>`, page 928.

HTML `lateximage`: Remembers and reuses the print version `\colorbox`.

`\fcolorbox`**:**

Print: Modified to allow a background of `none`.
`\LWR@print@fcolorbox` at section 81

HTML: Redefined by **lwarp-xcolor**, page 928.

HTML `lateximage`: Remembers and reuses the print version.

`\fcolorboxBlock`**:**

Print: Becomes `\fcolorbox`. Section 81

HTML: Newly defined by **lwarp-xcolor** to use a `<div>`, page 929.

HTML `lateximage`: Remembers and reuses the print version `\fcolorbox`.

`fcolorminipage`**:**

Print: Newly defined in the **lwarp** core.
`LWR@print@fcolorminipage` at section 81

HTML: Newly defined by **lwarp-xcolor**, page 930.

HTML `lateximage`: Uses the print version.

`\boxframe`**:**

Print: Used as-is.

HTML: Redefined by **lwarp-xcolor**, page 930.

HTML `lateximage`: Remembers and reuses the print version.

§ 409.3 Package loading

for HTML output:

```
\LWR@ProvidesPackagePass{xcolor}
\begin{warpHTML}
```

§ 409.4 Remembering and restoring original definitions

Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```
\LetLtxMacro\LWR@print@pagecolor\pagecolor
\LetLtxMacro\LWR@print@nopagecolor\nopagecolor
```

`\LWR@restoreorigformatting` Inside a `lateximage` the following gets restored to their print-mode actions:

```
\appto\LWR@restoreorigformatting{%
\LetLtxMacro\pagecolor\LWR@print@pagecolor%
\LetLtxMacro\nopagecolor\LWR@print@nopagecolor%
}
```

§ 409.5 HTML color style

`\LWR@findcurrenttextcolor` Sets `\LWR@tempcolor` to the current color.

```
\renewcommand*{\LWR@findcurrenttextcolor}{%
\protect\colorlet{LWR@current@color}{.}%
\protect\convertcolorspec{named}{LWR@current@color}{HTML}\LWR@tempcolor%
}
```

`\LWR@currenttextcolorstyle` Prints a color style for the current color.

```
\newcommand*{\LWR@currenttextcolorstyle}{%
\LWR@findcurrenttextcolor%
\ifdefstring{\LWR@tempcolor}{000000}%
{}%
{color: \LWR@origpound\LWR@tempcolor ; }%
}
```

`\LWR@textcurrentcolor` {⟨*text*⟩} Like `\textcolor` but uses the current `\color` instead.

```
\DeclareDocumentCommand{\LWR@textcurrentcolor}{m}{%
\begingroup%
\LWR@FBcancel%
\LWR@findcurrenttextcolor%
\InlineClass[color:\LWR@origpound\LWR@tempcolor]{textcolor}{%
    \renewcommand*{\LWR@currenttextcolor}{\LWR@origpound\LWR@tempcolor}%
    #1%
```

```
}%
\endgroup%
}
```

`\LWR@colorstyle` `{`⟨*2: model*⟩`}` `{`⟨*3: color*⟩`}`

For a color style, prints the color converted to HTML colors.

```
\NewDocumentCommand{\LWR@colorstyle}{m m}{%
\begingroup%
\LWR@FBcancel%
```

Use the **xcolor** package to convert to an HTML color space:

```
\convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Print the converted color:

```
\LWR@origpound\LWR@tempcolor%
\endgroup%
}
```

`\LWR@backgroundcolor` `[`⟨*model*⟩`]` `{`⟨*color*⟩`}` `{`⟨*text*⟩`}`

Similar to `\textcolor`, but prints black text against a color background.

Converted into an HTML hex color span.

```
\NewDocumentCommand{\LWR@backgroundcolor}{O{named} m m}{%
\begingroup%
\LWR@FBcancel%
\InlineClass[background:\LWR@colorstyle{#1}{#2}]{backgroundcolor}{%
#3%
}%
\endgroup%
}
```

§ 409.6 HTML border

`\LWR@borderpadding` `{`⟨*colorstyle*⟩`}` `{`⟨*color*⟩`}` Prints the HTML attributes for a black border and padding. `\LWR@forceminwidth` must be used first in order to set the border width.

```
\newcommand*{\LWR@borderpadding}[2]{%
border:\LWR@printlength{\LWR@atleastonepx} solid \LWR@colorstyle{#1}{#2} ; %
padding:\LWR@printlength{\fboxsep}%
}
```

§ 409.7 High-level macros

\textcolor `[⟨model⟩] {⟨color⟩} {⟨text⟩}`

Converted into an HTML hex color span.

```
\NewDocumentCommand{\LWR@HTML@textcolor}{o m m}{%
\begingroup%
```

Set the PDF color, to be picked up by SVG math if possible.

The print-mode `\color` command cannot accept the `named` option with color mixing, but it works with no option at all.

```
\IfValueTF{#1}{%
    \color[#1]{#2}%
}{%
    \color{#2}%
}%

\LWR@FBcancel%
\IfValueTF{#1}{%
\InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
\renewcommand*{\LWR@currenttextcolor}{\LWR@origpound\LWR@tempcolor}%
#3%
}%
}{%
\InlineClass[color:\LWR@colorstyle{named}{#2}]{textcolor}{%
\renewcommand*{\LWR@currenttextcolor}{\LWR@origpound\LWR@tempcolor}%
#3%
}%
}%
\endgroup%
}

\LWR@formatted{textcolor}
```

\pagecolor `[⟨model⟩] {⟨color⟩}`

Ignored. Use CSS instead.

```
\renewcommand*{\pagecolor}[2][named]{}
```

\nopagecolor Ignored.

```
\renewcommand*{\nopagecolor}{}
```

\colorbox [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Converted into an HTML hex background color .

```
\NewDocumentCommand{\LWR@HTML@colorbox}{O{named} m +m}{%
\begingroup%
\LWR@FBcancel%
\InlineClass[%
background:\LWR@colorstyle{#1}{#2} ; %
padding:\LWR@printlength{\fboxsep}%
]{colorbox}{#3}%
\endgroup%
}

\AtBeginDocument{
\LWR@formatted{colorbox}
}
```

\colorboxBlock [⟨*model*⟩] {⟨*color*⟩} {⟨*text*⟩}

Converted into an HTML hex background color .

```
\NewDocumentCommand{\LWR@HTML@colorboxBlock}{O{named} m +m}{%
\begingroup%
\LWR@FBcancel%
\begin{BlockClass}[%
background:\LWR@colorstyle{#1}{#2} ; %
padding:\LWR@printlength{\fboxsep}%
]{colorboxBlock}
#3
\end{BlockClass}%
\endgroup%
}

\AtBeginDocument{
\LWR@formatted{colorboxBlock}
}
```

\fcolorbox [⟨*framemodel*⟩] {⟨*framecolor*⟩} [⟨*boxmodel*⟩] {⟨*boxcolor*⟩} {⟨*text*⟩}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```
\NewDocumentCommand{\LWR@HTML@fcolorbox}{O{named} m O{named} m +m}{%
\LWR@traceinfo{HTML fcolorbox #2 #4}%
```

```
\begingroup%
\LWR@FBcancel%
\LWR@forceminwidth{\fboxrule}%
\ifthenelse{\equal{#4}{none}}%
{% no background color
    \InlineClass[%
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorbox}{#5}%
}%
{% yes background color
    \InlineClass[%
    \LWR@borderpadding{#1}{#2} ; %
    background:\LWR@colorstyle{#3}{#4}%
    ]{fcolorbox}{#5}%
}%
\endgroup%
}

\AtBeginDocument{
\LWR@formatted{fcolorbox}
}
```

`\fcolorboxBlock` `[`⟨*framemodel*⟩`]` `{`⟨*framecolor*⟩`}` `[`⟨*boxmodel*⟩`]` `{`⟨*boxcolor*⟩`}` `{`⟨*text*⟩`}`

Converted into a framed HTML hex background color span.

A background color of `none` creates a colored frame without a background color.

```
\NewDocumentCommand{\LWR@HTML@fcolorboxBlock}{O{named} m O{named} m +m}{%
\LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
\begingroup%
\LWR@FBcancel%
\LWR@forceminwidth{\fboxrule}%
\ifthenelse{\equal{#4}{none}}%
{% no background color
    \begin{BlockClass}[%
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorboxBlock}
    #5
    \end{BlockClass}%
}%
{% yes background color
    \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
    \begin{BlockClass}[%
    background:\LWR@origpound\LWR@tempcolortwo\ ; %
    \LWR@borderpadding{#1}{#2}%
    ]{fcolorboxBlock}
    #5
```

```
    \end{BlockClass}%
}%
\endgroup%
\LWR@traceinfo{HTML fcolorboxBlock done}%
}

\AtBeginDocument{
\LWR@formatted{fcolorboxBlock}
}
```

Creates a framed HTML around its contents.

A print-output version is defined in the **lwarp** core: section 81

\LWR@subfcolorminipage {⟨*framemodel*⟩} {⟨*framecolor*⟩} {⟨*background tag*⟩} {⟨*height*⟩}

```
\NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
\begin{BlockClass}[%
#3%
\LWR@borderpadding{#1}{#2} ; %
\IfValueT{#4}{height:\LWR@printlength{\LWR@tempheight} ; }%
width:\LWR@printlength{\LWR@tempwidth}%
]{fcolorminipage}%
}
```

Env fcolorminipage [⟨*1:framemodel*⟩] {⟨*2:framecolor*⟩} [⟨*3:boxmodel*⟩] {⟨*4:boxcolor*⟩} [⟨*5:align*⟩] [⟨*6:height*⟩] [⟨*7:inner-align*⟩] {⟨*8:width*⟩}

```
\NewDocumentEnvironment{LWR@HTML@fcolorminipage}{O{named} m O{named} m O{c} o o m}
{%
\LWR@FBcancel%
\setlength{\LWR@tempwidth}{#8}%
\IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
\LWR@forceminwidth{\fboxrule}%
\convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
\ifthenelse{\equal{#4}{none}}%
{\LWR@subfcolorminipage{#1}{#2}{}{#6}}%
{%
    \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
    \LWR@subfcolorminipage{#1}{#2}{background:\LWR@origpound\LWR@tempcolortwo\ ; }{#6}%
}%
}
{\end{BlockClass}}

\AtBeginDocument{
\LWR@formattedenv{fcolorminipage}
}
```

\boxframe {⟨*width*⟩} {⟨*height*⟩} {⟨*depth*⟩}

The depth is added to the height, but the box is not decended below by the depth. \textcolor is honored.

```
\newcommand*{\LWR@HTML@boxframe}[3]{%
{%
\setlength{\LWR@tempwidth}{#1}%
\setlength{\LWR@tempheight}{#2}%
\addtolength{\LWR@tempheight}{#3}%
\LWR@forceminwidth{\fboxrule}%
\LWR@findcurrenttextcolor%
\InlineClass[%
display:inline-block ; %
border:\LWR@printlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
width:\LWR@printlength{\LWR@tempwidth} ; %
height:\LWR@printlength{\LWR@tempheight}%
]{boxframe}{}%
}%
}

\LWR@formatted{boxframe}
```

§ 409.8 Row colors

\rowc@l@rs [⟨*cmds*⟩] {⟨*startrow*⟩} {⟨*odd color*⟩} {⟨*even color*⟩}

```
\newcommand*{\LWR@xcolortempcolor}{}

\def\rowc@l@rs[#1]#2#3#4%
{
\global\rownum=1
   \global\@rowcolorstrue
   \@ifxempty{#3}%
     {\def\@oddrowcolor{\@norowcolor}}%
     {%
        \convertcolorspec{named}{#3}{HTML}\LWR@xcolortempcolor%
        \edef\@oddrowcolor{%
            \csdef{LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
        }%
     }%
   \@ifxempty{#4}%
     {\def\@evenrowcolor{\@norowcolor}}%
     {%
        \convertcolorspec{named}{#4}{HTML}\LWR@xcolortempcolor%
        \edef\@evenrowcolor{%
            \csdef{LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
        }%
     }%
   \if@rowcmd
```

```
    \def\@rowcolors
    {%
%         #1%
      \if@rowcolors
%       \noalign{%
          \relax\ifnum\rownum<#2\@norowcolor\else
          \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi\fi%
%        }%
      \fi%
    }%
  \else
    \def\@rowcolors
    {%
      \if@rowcolors
          \ifnum\rownum<#2%
%          \noalign{%
              \@norowcolor
%           }
          \else
%           #1%
%           \noalign{%
              \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi%
%           }%
          \fi
      \fi%
    }%
  \fi
  \ignorespaces%
}
```

\@norowcolor Turns off color for this row.

```
\def\@norowcolor{%
\renewcommand{\LWR@xcolorrowHTMLcolor}{}%
}
```

\@rowc@lors Executed at the end of each row.

```
\def\@rowc@lors{%
%   \noalign{%
      \global\advance\rownum\@ne%
%   }%
    \@rowcolors%
}


\end{warpHTML}
```

File 318 `lwarp-xellipsis.sty`

§ 410 Package xellipsis

(*Emulates or patches code by* Donald P. Goodman III.)

Pkg `xellipsis` **xellipsis** is patched for use by **lwarp**.

When non-zero, each of the spaces is converted to an HTML thin unbreakable space.

for HTML output:

```
\LWR@ProvidesPackagePass{xellipsis}

\newcommand*{\LWR@xellipsespace}[1]{%
\ifdim#1=0pt\else%
    \ifdim#1<\fontdimen2\font%
        \,%
    \else%
        ~%
    \fi%
\fi%
}

\def\xelip{%
\mbox{%
    \LWR@xellipsespace{\xelipprebef}%
    \xelipprechar%
    \LWR@xellipsespace{\xelippreaft}%
    \LWR@xellipsespace{\xelipbef}%
    \xelipchar%
    \xel@loopi = 1%
    \loop\ifnum\xelipnum>\xel@loopi%
        \advance\xel@loopi by1%
        \LWR@xellipsespace{\xelipgap}%
        \xelipchar%
    \repeat%
    \LWR@xellipsespace{\xelipaft}%
    \LWR@xellipsespace{\xelippostbef}%
    \xelippostchar%
    \LWR@xellipsespace{\xelippostaft}%
}%
}%
```

File 319 `lwarp-xfrac.sty`

§ 411 Package

xfrac

(*Emulates or patches code by* The LaTeX3 Project.)

Pkg `xfrac` Supported by adding **xfrac** instances.

for HTML output:

```
1 \LWR@ProvidesPackagePass{xfrac}
```

⚠ font size In the user's document preamble, **lwarp** should be loaded after font-related setup. During HTML conversion, this font is used by **lwarp** to generate its initial PDF output containing HTML tags, later to be converted by **pdftotext** to a plain text file. While the text may be in any font which **pdftotext** can read, the math is directly converted into SVG images using this same user-selected font. `xfrac` below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine `\xfracHTMLfontsize` with a different em size.

`\sfrac` `[`⟨*instance*⟩`] {`⟨*num*⟩`} [`⟨*sep*⟩`] {`⟨*denom*⟩`}`

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. `\scalebox` is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

for HTML & PRINT:

```
2 \begin{warpall}
```

`\xfracHTMLfontsize` User-redefinable macro which controls the font size of the fraction.

```
3 \newcommand*{\xfracHTMLfontsize}{.6em}
```

```
4 \end{warpall}
```

for HTML output:

```
5 \begin{warpHTML}
```

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*{\LWR@htmlsmallfontstart}{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}%
8 \LWR@nestspan%
9 %
10 }
```

```
\newcommand*{\LWR@htmlsmallfontend}{%
\LWR@htmltagc{/span}%
\endLWR@nestspan%
}
```

instances Instances of **xfrac** for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that **pdftotext** correctly reads the result.

```
\DeclareInstance{xfrac}{default}{text}{
numerator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
\endgroup%
},
denominator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend%
\endgroup%
},
```

For **pdftotext**, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmr}{text}{
numerator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
\endgroup%
},
denominator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend%
\endgroup%
},
```

For **pdftotext**, do not scale the text:

```
scaling = false
```

```
}

\DeclareInstance{xfrac}{lmss}{text}{
numerator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
\endgroup%
},
denominator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend%
\endgroup%
},
```

For **pdftotext**, do not scale the text:

```
scaling = false
}

\DeclareInstance{xfrac}{lmtt}{text}{
numerator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
\endgroup%
},
denominator-format = {%
\begingroup%
\RenewDocumentCommand{\scalebox}{m o m}{##3}%
\LWR@htmlsmallfontstart{}\,#1\LWR@htmlsmallfontend%
\endgroup%
},
```

For **pdftotext**, do not scale the text:

```
scaling = false
}



\end{warpHTML}
```

File 320 `lwarp-xltabular.sty`

§ 412 Package xltabular

(*Emulates or patches code by* Rolf Niepraschk, Herbert Voss.)

Pkg `xltabular` **xltabular** is emulated by **lwarp**.

for HTML output: Relies on **tabularx**.

⚠ table numbering At present, an `xltabular` without a caption or with only a `\caption*` may be misnumbered in HTML, so it may be necessary to place at the end of the table:

```
\warpHTMLonly{\addtocounter{table}{-1}}
```

```
\RequirePackage{tabularx}

\LWR@ProvidesPackageDrop{xltabular}

\DeclareDocumentEnvironment{xltabular}{o m m}
{\longtable{#3}}
{\endlongtable}
```

File 321 `lwarp-xltxtra.sty`

§ 413 Package xltxtra

(*Emulates or patches code by* Will Robertson, Jonathan Kew.)

Pkg `xltxtra` **xltxtra** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{xltxtra}

\RequirePackage{realscripts}
\RequirePackage{metalogo}
\newcommand*\TeX@logo@spacing[6]{}

\newcommand*{\vfrac}[2]{%
\textsuperscript{#1}/\textsubscript{#2}%
}

\newcommand\namedglyph[1]{%
  \@tempcnta=\XeTeXglyphindex "#1"\relax
```

```
  \ifnum\@tempcnta>0
    \XeTeXglyph\@tempcnta
  \else
    \xxt@namedglyph@fallback{#1}%
  \fi}

\newcommand\xxt@namedglyph@fallback[1]{[#1]}

\DeclareDocumentCommand{\showhyphens}{m}{}
```

File 322 `lwarp-xmpincl.sty`

§ 414 Package xmpincl

(*Emulates or patches code by* Maarten Sneep.)

Pkg `xmpincl` Emulated.

for HTML output: Discard all options for **lwarp-xmpincl**:

```
\LWR@ProvidesPackageDrop{xmpincl}

\newcommand*{\includexmp}[1]{}
```

File 323 `lwarp-xpiano.sty`

§ 415 Package xpiano

(*Emulates or patches code by* Enrico Gregorio.)

Pkg `xpiano` **xpiano** is patched for use by **lwarp**.

for HTML output:

```
\LWR@ProvidesPackagePass{xpiano}

\ExplSyntaxOn
\NewDocumentCommand{\LWR@print@keyboard}{ O{}m }
{
\xpiano_keyboard:nn { #1 } { #2 }
}

\NewDocumentCommand{\LWR@HTML@keyboard}{ O{}m }
{
\begin{lateximage}*
    [-xpiano-~\packagediagramname{}: \detokenize\expandafter{#2}]
```

```
    [\detokenize\expandafter{#1}]
\xpiano_keyboard:nn { #1 } { #2 }
\end{lateximage}
}
\ExplSyntaxOff

\LWR@formatted{keyboard}
```

File 324 `lwarp-xtab.sty`

§ 416 Package

xtab

(*Emulates or patches code by* Peter Wilson.)

Pkg xtab **xtab** is emulated.

for HTML output:

```
\LWR@ProvidesPackageDrop{xtab}
```

⚠ misplaced alignment
alignment tab character &

For `\tablefirsthead`, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\StopDefiningTabulars
```

See section 9.9.

⚠ lateximage **supertabular** and **xtab** are not supported inside a `lateximage`.

```
\newcommand{\LWRXT@firsthead}{}

\newcommand{\tablefirsthead}[1]{%
    \long\gdef\LWRXT@firsthead{#1}%
}

\newcommand{\tablehead}[1]{}

\newcommand{\tablelasthead}[1]{}

\newcommand{\notablelasthead}{}

\newcommand{\tabletail}[1]{}

\newcommand{\LWRXT@lasttail}{}

\newcommand{\tablelasttail}[1]{%
    \long\gdef\LWRXT@lasttail{#1}%
```

```
}

\newcommand{\tablecaption}[2][]{%
    \long\gdef\LWRXT@caption{\caption[#1]{#2}}%
}

\let\topcaption\tablecaption
\let\bottomcaption\tablecaption


\newcommand*{\LWRXT@caption}{}

\newcommand*{\shrinkheight}[1]{}

\newcommand*{\xentrystretch}[1]{}

\NewDocumentEnvironment{xtabular}{s o m}
{%
\LWR@traceinfo{xtabular}%
\table%
\LWRXT@caption%
\begin{tabular}{#3}%
\TabularMacro\ifdefvoid{\LWRXT@firsthead}%
{\LWR@getmynexttoken}%
{\expandafter\LWR@getmynexttoken\LWRXT@firsthead}%
}%
{%
\ifdefvoid{\LWRXT@lasttail}%
{}%
{%
\TabularMacro\ResumeTabular%
\LWRXT@lasttail%
}%
\end{tabular}%
\endtable%
\LWR@traceinfo{xtabular done}%
}

\NewDocumentEnvironment{mpxtabular}{s o m}
{\minipage{\linewidth}\xtabular{#3}}
{\endxtabular\endminipage}
```

File 325 `lwarp-xunicode.sty`

§ 417 Package xunicode

Pkg `xunicode` Error if **xunicode** is loaded after **lwarp**.

Patch **lwarp-xunicode**, but also verify that is was loaded before **lwarp**:

for HTML output:

```
\@ifpackageloaded{lwarp}{\LWR@loadbefore{xunicode}}{}

\LWR@ProvidesPackagePass{xunicode}[2011/09/09]
```

`\textcircled` becomes a span with a rounded border:

```
\newcommand*{\LWR@HTML@textcircled}[1]{%
\InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{#1}%
}

\LWR@formatted{textcircled}
```

File 326 `lwarp-xurl.sty`

§ 418 Package xurl

Pkg `xurl` **xurl** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{xurl}
```

File 327 `lwarp-xy.sty`

§ 419 Package xy

(*Emulates or patches code by* Kristoffer H. Rose, Ross Moore.)

Pkg `xy` **xy** is patched for use by **lwarp**.

⚠ `\xypolygon` must be used inside the `xy` environment, or inside `\xy` … `\endxy`.

for HTML output:

```
\LWR@ProvidesPackagePass{xy}
```

```
\AtBeginDocument{

\preto{\xy}{\begin{lateximage}[(-xy--\packagediagramname)]}
\appto{\endxy}{\end{lateximage}}

\@ifundefined{xymatrix}{}{
\LetLtxMacro\LWR@origxymatrix\xymatrix

\renewcommand{\xymatrix}[1]{%
\begin{lateximage}[(-xy- xymatrix \packagediagramname)]
\LWR@origxymatrix{#1}
\end{lateximage}
}
}

\@ifundefined{xygraph}{}{
\LetLtxMacro\LWR@origxygraph\xygraph

\renewcommand{\xygraph}[1]{%
\begin{lateximage}[(-xy- xygraph \packagediagramname)]
\LWR@origxygraph{#1}
\end{lateximage}
}
}

}
```

File 328 `lwarp-zwpagelayout.sty`

§ 420 Package zwpagelayout

(*Emulates or patches code by* Zdeněk Wagner.)

Pkg `zwpagelayout` **zwpagelayout** is ignored.

for HTML output:

```
\LWR@ProvidesPackageDrop{zwpagelayout}

\def\noBboxes{}
\@onlypreamble\noBboxes

\expandafter\ifx\csname definecolor\endcsname\relax \else
  \definecolor{cmykblack}{cmyk}{0,0,0,1}
  \definecolor{grblack}{gray}{0}
%   \ifzwpl@redefineblack
%     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
%   \fi
  \definecolor{cmykred}{cmyk}{0,1,1,0}
```

```
  \definecolor{cmykgreen}{cmyk}{1,0,1,0}
  \definecolor{cmykblue}{cmyk}{1,1,0,0}
  \definecolor{rgbred}{rgb}{1,0,0}
  \definecolor{rgbgreen}{rgb}{0,1,0}
  \definecolor{rgbblue}{rgb}{0,0,1}
%   \ifzwpl@redefinetocmyk
%     \definecolor{red}{cmyk}{0,1,1,0}
%     \definecolor{green}{cmyk}{1,0,1,0}
%     \definecolor{blue}{cmyk}{1,1,0,0}
%   \fi
\fi

\let\OverprintXeTeXExtGState\relax

\DeclareRobustCommand\SetOverprint{\ignorespaces}
\DeclareRobustCommand\SetKnockout{\ignorespaces}
\DeclareRobustCommand\textoverprint[1]{{\SetOverprint#1}}
\DeclareRobustCommand\textknockout[1]{{\SetKnockout#1}}

\def\SetPDFminorversion#1{}
\@onlypreamble\SetPDFminorversion

\newcommand*\Vcorr{}

\DeclareRobustCommand\vb[1][]{}
\NewDocumentCommand{\NewOddPage}{* o}{}
\NewDocumentCommand{\NewEvenPage}{* o}{}
\def\SetOddPageMessage#{\gdef\ZW@oddwarning}
\def\SetEvenPageMessage#{\gdef\Z@@evenwarning}
\def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning

\def\clap#1{#1}

\def\CropFlap{2in}
\def\CropSpine{1in}
\def\CropXSpine{1in}
\def\CropXtrim{.25in}
\def\CropYtrim{.25in}
\def\UserWidth{5in}
\def\UserLeftMargin{1in}
\def\UserRightMargin{1in}
\def\UserTopMargin{1in}
\def\UserBotMargin{1in}
\def\thePageNumber{\LWR@origpound\,\arabic{page}}
\ifXeTeX
\def\ifcaseZWdriver{\ifcase2}
\else
\def\ifcaseZWdriver{\ifcase1}
\fi
\DeclareRobustCommand\ZWifdriver[2]{}
```

File 329 `lwarp-patch-komascript.sty`

§ 421 Package

patch-komascript

Pkg `lwarp-patch-komascript`

Patches for **komascript** classes.

lwarp loads this package when **scrbook**, **scrartcl**, or **scrreprt** classes are detected.

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

`\titlehead`, `\subject`, `\captionformat`, `\figureformat`, and `\tableformat` are not yet emulated.

⚠ Not fully tested! Please send bug reports!

Some features have not yet been tested. Please contact the author with any bug reports.

for HTML output:

```
\ProvidesPackage{lwarp-patch-komascript}
```

typearea is emulated.

```
\RequirePackage{lwarp-typearea}
```

tocbasic is emulated.

```
\RequirePackage{lwarp-tocbasic}
```

scrextend patches most of the new macros.

```
\RequirePackage{lwarp-scrextend}
```

Indexing macros, simplified for **lwarp**:

```
\AtBeginDocument{

\renewcommand*{\idx@heading}{%
  \idx@@heading{\indexname}%
}

\renewenvironment{theindex}{%
  \idx@heading%
  \index@preamble\par\nobreak
```

```
    \let\item\LWR@indexitem%
    \let\subitem\LWR@indexsubitem%
    \let\subsubitem\LWR@indexsubsubitem%
}
{}

\renewcommand*\indexspace{}

}% AtBeginDocument
```

The \minisec is placed inside a of class minisec.

```
\renewcommand*{\minisec}[1]{
\begin{BlockClass}{minisec}
#1
\end{BlockClass}
}
```

The part and chapter preambles are placed as plain text just after each heading.

```
\@ifundefined{setpartpreamble}{}{
\RenewDocumentCommand{\setpartpreamble}{o o +m}{%
\renewcommand{\part@preamble}{#3}%
}
}

\@ifundefined{setchapterpreamble}{}{
\RenewDocumentCommand{\setchapterpreamble}{o o +m}{%
\renewcommand{\chapter@preamble}{#3}%
}
}
```

Simple captions are used in all cases.

```
\LetLtxMacro\captionbelow\caption
\LetLtxMacro\captionabove\caption

\LetLtxMacro\captionofbelow\captionof
\LetLtxMacro\captionofabove\captionof

\RenewDocumentEnvironment{captionbeside}{o m o o o s}
{}
{%
\IfValueTF{#1}%
{\caption[#1]{#2}}%
{\caption{#2}}%
}

\RenewDocumentEnvironment{captionofbeside}{m o m o o o s}
```

```
{}
{%
\IfValueTF{#2}%
{\captionof{#1}[#2]{#3}}%
{\captionof{#1}{#3}}%
}

\RenewDocumentCommand{\setcapindent}{s m}{}
\renewcommand*{\setcaphanging}{}
\renewcommand*{\setcapwidth}[2][]{}
\renewcommand*{\setcapdynwidth}[2][]{}
\RenewDocumentCommand{\setcapmargin}{s o m}{}
```

File 330 `lwarp-patch-memoir.sty`

§ 422 Package

patch-memoir

(*Emulates or patches code by* Peter Wilson.)

Pkg `lwarp-patch-memoir` Patches for **memoir** class.

⚠ Not fully tested! Please send bug reports!

lwarp loads this package when the **memoir** class is detected.

⚠ options clash

While emulating **memoir**, **lwarp** pre-loads a number of packages (section 422.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading **lwarp**:

```
\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}
```

`\verbfootnote` is not supported.

`\newfootnoteseries`, etc. are not supported.

lwarp loads **pagenote** to perform **memoir**'s pagenote functions, but there are minor differences in `\pagenotesubhead` and related macros.

Poem numbering is not supported.

The `verbatim` environment does not yet support the **memoir** enhancements. It is currently recommended to load and use **fancyvrb** instead.

The **memoir** glossary system is not yet supported by **lwarpmk**. The **glossaries** package may be used instead, but does require the glossary entries be changed from the **memoir** syntax to the **glossaries** syntax.

for HTML output:

```
\ProvidesPackage{lwarp-patch-memoir}
```

§ 422.1 Packages

These are pre-loaded to provide emulation for many of **memoir**'s functions. **memoir** pretends that **abstract**, etc. are already loaded, via its "emulated" package mechanism, but **lwarp** is directly loading the "`lwarp-`" version of each, which happens to avoid **memoir**'s emulation system.

```
\RequirePackage{lwarp-abstract}% req'd
\RequirePackage{lwarp-array}% req'd
\RequirePackage{lwarp-booktabs}% req'd
% \RequirePackage{lwarp-ccaption}% emualated below
\RequirePackage{lwarp-changepage}% req'd
\RequirePackage{lwarp-crop}
\RequirePackage{lwarp-dcolumn}% req'd
\RequirePackage{lwarp-enumerate}% req'd
\RequirePackage{lwarp-epigraph}% req'd
\RequirePackage{lwarp-fancyvrb}% req'd
\RequirePackage{lwarp-footmisc}% req'd
\RequirePackage{lwarp-framed}% req'd
\RequirePackage{lwarp-hanging}% req'd
\RequirePackage{lwarp-makeidx}% req'd
\DisemulatePackage{moreverb}
\RequirePackage{lwarp-moreverb}
\RequirePackage{lwarp-mparhack}
\RequirePackage{lwarp-needspace}% req'd
\RequirePackage{lwarp-nextpage}% req'd
\RequirePackage{lwarp-pagenote}% req'd
\RequirePackage{lwarp-parskip}
\RequirePackage{lwarp-setspace}% req'd
\RequirePackage{lwarp-showidx}
\RequirePackage{lwarp-subfigure}% red'q
\makeindex
```

subfigure is emulated via **subfig**, which pre-defines `subfigure` and `subtable`, but **memoir** does not, so they must be tested for here:

```
\LetLtxMacro\LWR@memorignewsubfloat\newsubfloat
\RenewDocumentCommand{\newsubfloat}{O{} m}{%
    \@ifundefined{c@sub#2}{%
        \LWR@memorignewsubfloat[#1]{#2}%
```

```
    }{}%
}

\RequirePackage{lwarp-tabularx}% req'd
\RequirePackage{lwarp-titling}% req'd
% \RequirePackage{lwarp-tocbibind}% not emulated by memoir
\RequirePackage{lwarp-tocloft}% req'd
\RequirePackage{lwarp-verse}% req'd
```

§ 422.2 Preliminary setup

Bypass the **memoir** package mechanism:

```
\LetLtxMacro\LWR@orig@label\@mem@old@label
```

memoir already set the page size to a default, so it must be forced large for **lwarp**'s use, to avoid tag overflows off the page.

```
\setstocksize{190in}{20in}
\setlrmarginsandblock{2in}{2in}{*}
\setulmarginsandblock{1in}{1in}{*}
```

§ 422.3 Laying out the page

```
\renewcommand*{\stockavi}{}
\renewcommand*{\stockav}{}
\renewcommand*{\stockaiv}{}
\renewcommand*{\stockaiii}{}
\renewcommand*{\stockbvi}{}
\renewcommand*{\stockbv}{}
\renewcommand*{\stockbiv}{}
\renewcommand*{\stockbiii}{}
% \renewcommand*{\stockmetriccrownvo}{}% in docs but not in the package
\renewcommand*{\stockmlargecrownvo}{}
\renewcommand*{\stockmdemyvo}{}
\renewcommand*{\stockmsmallroyalvo}{}
\renewcommand*{\pageavi}{}
\renewcommand*{\pageav}{}
\renewcommand*{\pageaiv}{}
\renewcommand*{\pageaiii}{}
\renewcommand*{\pagebvi}{}
\renewcommand*{\pagebv}{}
\renewcommand*{\pagebiv}{}
\renewcommand*{\pagebiii}{}
% \renewcommand*{\pagemetriccrownvo}{}% in docs but not in the package
\renewcommand*{\pagemlargecrownvo}{}
\renewcommand*{\pagemdemyvo}{}
\renewcommand*{\pagemsmallroyalvo}{}

```

```
\renewcommand*{\stockdbill}{}
\renewcommand*{\stockstatement}{}
\renewcommand*{\stockexecutive}{}
\renewcommand*{\stockletter}{}
\renewcommand*{\stockold}{}
\renewcommand*{\stocklegal}{}
\renewcommand*{\stockledger}{}
\renewcommand*{\stockbroadsheet}{}
\renewcommand*{\pagedbill}{}
\renewcommand*{\pagestatement}{}
\renewcommand*{\pageexecutive}{}
\renewcommand*{\pageletter}{}
\renewcommand*{\pageold}{}
\renewcommand*{\pagelegal}{}
\renewcommand*{\pageledger}{}
\renewcommand*{\pagebroadsheet}{}

\renewcommand*{\stockpottvo}{}
\renewcommand*{\stockfoolscapvo}{}
\renewcommand*{\stockcrownvo}{}
\renewcommand*{\stockpostvo}{}
\renewcommand*{\stocklargecrownvo}{}
\renewcommand*{\stocklargepostvo}{}
\renewcommand*{\stocksmalldemyvo}{}
\renewcommand*{\stockdemyvo}{}
\renewcommand*{\stockmediumvo}{}
\renewcommand*{\stocksmallroyalvo}{}
\renewcommand*{\stockroyalvo}{}
\renewcommand*{\stocksuperroyalvo}{}
\renewcommand*{\stockimperialvo}{}
\renewcommand*{\pagepottvo}{}
\renewcommand*{\pagefoolscapvo}{}
\renewcommand*{\pagecrownvo}{}
\renewcommand*{\pagepostvo}{}
\renewcommand*{\pagelargecrownvo}{}
\renewcommand*{\pagelargepostvo}{}
\renewcommand*{\pagesmalldemyvo}{}
\renewcommand*{\pagedemyvo}{}
\renewcommand*{\pagemediumvo}{}
\renewcommand*{\pagesmallroyalvo}{}
\renewcommand*{\pageroyalvo}{}
\renewcommand*{\pagesuperroyalvo}{}
\renewcommand*{\pageimperialvo}{}

\renewcommand*{\memfontfamily}{}
\renewcommand*{\memfontenc}{}
\renewcommand*{\memfontpack}{}

\renewcommand*{\anyptfilebase}{}
\renewcommand*{\anyptsize}{10}
```

```
118
119 \renewcommand*{\setstocksize}[2]{}
120 \renewcommand*{\settrimmedsize}[3]{}
121 \renewcommand*{\settrims}[2]{}
122
123 % \newlength{\lxvchars}
124 % \setlength{\lxvchars}{305pt}
125 % \newlength{\xlvchars}
126 % \setlength{\xlvchars}{190pt}
127 \renewcommand*{\setxlvchars}[1]{}
128 \renewcommand*{\setlxvchars}[1]{}
129
130 \renewcommand*{\settypeblocksize}[3]{}
131 \renewcommand*{\setlrmargins}[3]{}
132 \renewcommand*{\setlrmarginsandblock}[3]{}
133 \renewcommand*{\setbinding}[1]{}
134 \renewcommand*{\setulmargins}[3]{}
135 \renewcommand*{\setulmarginsandblock}[3]{}
136 \renewcommand*{\setcolsepandrule}[2]{}
137
138 \renewcommand*{\setheadfoot}[2]{}
139 \renewcommand*{\setheaderspaces}[3]{}
140 \renewcommand*{\setmarginnotes}[3]{}
141 \renewcommand*{\setfootins}[2]{}
142 \renewcommand*{\checkandfixthelayout}[1][]{}
143 \renewcommand*{\checkthelayout}[1]{}
144 \renewcommand*{\fixthelayout}{}
145 %
146 % \newlength{\stockheight}
147 % \newlength{\trimtop}
148 % \newlength{\trimedge}
149 % \newlength{\stockwidth}
150 % \newlength{\spinemargin}
151 % \newlength{\foremargin}
152 % \newlength{\uppermargin}
153 % \newlength{\headmargin}
154 %
155 \renewcommand*{\typeoutlayout}{}
156 \renewcommand*{\typeoutstandardlayout}{}
157 \renewcommand*{\settypeoutlayoutunit}[1]{}
158 \renewcommand*{\fixpdflayout}{}
159 \renewcommand*{\fixdvipslayout}{}
160
161 \renewcommand*{\medievalpage}[1][]{}
162 \renewcommand*{\isopage}[1][]{}
163 \renewcommand*{\semiisopage}[1][]{}
164
165 \renewcommand{\setpagebl}[3]{}
166 \renewcommand{\setpageml}[3]{}
167 \renewcommand{\setpagetl}[3]{}
```

```
\renewcommand{\setpagetm}[3]{}
\renewcommand{\setpagetr}[3]{}
\renewcommand{\setpagemr}[3]{}
\renewcommand{\setpagebr}[3]{}
\renewcommand{\setpagebm}[3]{}
\renewcommand{\setpagecc}[3]{}
```

§ 422.4 Text and fonts

```
\let\miniscule\tiny
\let\HUGE\Huge

\renewcommand*{\abnormalparskip}[1]{}
\renewcommand*{\nonzeroparskip}{}
\renewcommand*{\traditionalparskip}{}

\let\onelineskip\baselineskip

\let\OnehalfSpacing\onehalfspacing
\let\DoubleSpacing\doublespacing
\renewcommand*{\setPagenoteSpacing}[1]{}
\renewcommand*{\setFloatSpacing}[1]{}
\let\SingleSpacing\singlespacing
\let\setSingleSpace\SetSinglespace
\let\SingleSpace\singlespace
\let\endSingleSpace\endsinglespace
\let\Spacing\spacing
\let\endSpacing\endspacing
\let\OnehalfSpace\onehalfspace
\let\endOnehalfSpace\endonehalfspace
\csletcs{OnehalfSpace*}{onehalfspace}
\csletcs{endOnehalfSpace*}{endonehalfspace}
\let\DoubleSpace\doublespace
\let\endDoubleSpace\enddoublespace
\csletcs{DoubleSpace*}{doublespace}
\csletcs{endDoubleSpace*}{enddoublespace}
\renewcommand*{\setDisplayskipStretch}[1]{}
\renewcommand*{\memdskipstretch}{}
\renewcommand*{\noDisplayskipStretch}{}
\renewcommand*{\memdskips}{}

\renewcommand*{\midsloppy}{}
\renewenvironment*{midsloppypar}{}{}

\renewcommand*{\sloppybottom}{}
```

§ 422.5 Titles

```
\csletcs{titlingpage*}{titlingpage}
```

```
\csletcs{endtitlingpage*}{endtitlingpage}
\let\titlingpageend\relax
\newcommand{\titlingpageend}[2]{}
\let\andnext\and
\renewcommand*{\thanksmarkstyle}[1]{}
\renewcommand{\thanksfootmark}{%
\thanksscript{\tamark}%
}

% \newlength{\thanksmarksep}
```

§ 422.6 Abstracts

```
\renewcommand*{\abstractcol}{}
\renewcommand*{\abstractintoc}{}
\renewcommand*{\abstractnum}{}
\renewcommand*{\abstractrunin}{}
```

§ 422.7 Docment divisions

```

\def\@apppage{%
    \part*{\appendixpagename}
}
\renewcommand\mempreaddapppagetotochook{}
\renewcommand\mempostaddapppagetotochook{}

\def\@sapppage{%
    \part*{\appendixpagename}
}

\csletcs{frontmatter*}{frontmatter}
\csletcs{mainmatter*}{mainmatter}
\renewcommand*{\raggedbottomsection}{}
\renewcommand*{\normalbottomsection}{}
\renewcommand*{\bottomsectionskip}{}
\renewcommand*{\bottomsectionpenalty}{}
\csletcs{appendixpage*}{appendixpage}
\renewcommand*{\namedsubappendices}{}
\renewcommand*{\unnamedsubappendices}{}
\renewcommand*{\setsecnumdepth}[1]{}% todo tocvsec2
\renewcommand*{\maxsecnumdepth}[1]{}% todo tocvsec2
\renewcommand*{\beforebookskip}{}
\renewcommand*{\afterbookskip}{}
\renewcommand*{\beforepartskip}{}
\renewcommand*{\afterpartskip}{}
\renewcommand*{\midbookskip}{}
\renewcommand*{\midpartskip}{}
\renewcommand*{\printbookname}{}
```

```
\renewcommand*{\booknamefont}{}
\renewcommand*{\booknamenum}{}
\renewcommand*{\printbooknum}{}
\renewcommand*{\booknumfont}{}
\renewcommand*{\printpartname}{}
\renewcommand*{\partnamefont}{}
\renewcommand*{\partnamenum}{}
\renewcommand*{\printpartnum}{}
\renewcommand*{\partnumfont}{}
\renewcommand*{\printbooktitle}[1]{}
\renewcommand*{\booktitlefont}{}
\renewcommand{\printparttitle}[1]{}
\renewcommand*{\parttitlefont}{}
\renewcommand*{\bookpageend}{}
\renewcommand*{\bookblankpage}{}
\renewcommand*{\nobookblankpage}{}
\renewcommand*{\partpageend}{}
\renewcommand*{\partblankpage}{}
\renewcommand*{\nopartblankpage}{}
\RenewDocumentCommand{\newleadpage}{s o m m}{}% todo
\RenewDocumentCommand{\renewleadpage}{s o m m}{}% todo
\renewcommand*{\leadpagetoclevel}{chapter}

\renewcommand*{\openright}{}
\renewcommand*{\openleft}{}
\renewcommand*{\openany}{}
\renewcommand*{\clearforchapter}{}
\renewcommand*{\memendofchapterhook}{}
\renewcommand*{\chapterheadstart}{}
% \newlength{\beforechapskip}
\renewcommand*{\afterchapternum}{}
% \newlength{\midchapskip}
\renewcommand*{\afterchaptertitle}{}
% \newlength{\afterchapskip}
\renewcommand*{\printchaptername}{}
\renewcommand*{\chapnamefont}{}
\renewcommand*{\chapternamenum}{}
\renewcommand*{\printchapternum}{}
\renewcommand*{\chapnumfont}{}
\renewcommand{\printchaptertitle}[1]{}
\renewcommand*{\chaptitlefont}{}
\renewcommand*{\printchapternonum}{}
\renewcommand*{\indentafterchapter}{}
\renewcommand*{\noindentafterchapter}{}
\renewcommand*{\insertchapterspace}{}

\renewcommand*{\chapterstyle}[1]{}
\renewcommand{\makechapterstyle}[2]{}
\renewcommand*{\chapindent}{}
\let\chapterprecis\cftchapterprecis
```

```
\let\chapterprecishere\cftchapterprecishere
\let\chapterprecistoc\cftchapterprecistoc
\renewcommand*{\precisfont}{}
\renewcommand*{\prechapterprecis}{}
\renewcommand*{\postchapterprecis}{}
\renewcommand{\precistoctext}[1]{}
\renewcommand*{\precistocfont}{}
\renewcommand*{\precistocformat}{}
% \newlength{\prechapterprecisshift}

\renewcommand*{\setbeforesecskip}[1]{}
\renewcommand*{\setaftersecskip}[1]{}
\renewcommand*{\setsecindent}[1]{}
\renewcommand*{\setsecheadstyle}[1]{}
\renewcommand*{\setbeforesubsecskip}[1]{}
\renewcommand*{\setaftersubsecskip}[1]{}
\renewcommand*{\setsubsecindent}[1]{}
\renewcommand*{\setsubsecheadstyle}[1]{}
\renewcommand*{\setbeforesubsubsecskip}[1]{}
\renewcommand*{\setaftersubsubsecskip}[1]{}
\renewcommand*{\setsubsubsecindent}[1]{}
\renewcommand*{\setsubsubsecheadstyle}[1]{}
\renewcommand*{\setbeforeparaskip}[1]{}
\renewcommand*{\setafterparaskip}[1]{}
\renewcommand*{\setparaindent}[1]{}
\renewcommand*{\setparaheadstyle}[1]{}
\renewcommand*{\setbeforesubparaskip}[1]{}
\renewcommand*{\setaftersubparaskip}[1]{}
\renewcommand*{\setsubparaindent}[1]{}
\renewcommand*{\setsubparaheadstyle}[1]{}
\renewcommand{\@hangfrom}[1]{#1}
\renewcommand{\sethangfrom}[1]{}
\renewcommand{\setsecnumformat}[1]{}

\renewcommand*{\hangsecnum}{}
\renewcommand*{\defaultsecnum}{}

\renewcommand*{\sechook}{}
\renewcommand{\setsechook}[1]{}
\renewcommand*{\subsechook}{}
\renewcommand{\setsubsechook}[1]{}
\renewcommand*{\subsubsechook}{}
\renewcommand{\setsubsubsechook}[1]{}
\renewcommand*{\parahook}{}
\renewcommand{\setparahook}[1]{}
\renewcommand*{\subparahook}{}
\renewcommand{\setsubparahook}[1]{}

\RenewDocumentCommand{\plainbreak}{s m}{\begin{center}~\end{center}}

```

```
\RenewDocumentCommand{\fancybreak}{s +m}{%
\begin{center}#2\end{center}%
}

\RenewDocumentCommand{\plainfancybreak}{s m m +m}{%
\begin{center}#4\end{center}%
}

\RenewDocumentCommand{\pfbreak}{s}{%
\begin{center}
\pfbreakdisplay
\end{center}
}

% \newlength{\pfbreakskip}
\renewcommand{\pfbreakdisplay}{*\quad*\quad*}

\renewcommand{\makeheadstyles}[2]{}
\renewcommand*{\headstyles}[1]{}
```

§ 422.8 Pagination and headers

```
\renewcommand*{\savepagenumber}{}
\renewcommand*{\restorepagenumber}{}
\renewcommand*{\uppercaseheads}{}
\renewcommand*{\nouppercaseheads}{}

\renewcommand*{\bookpagemark}[1]{}
\renewcommand*{\partmark}[1]{}
\renewcommand*{\bibmark}{}
\renewcommand*{\indexmark}{}
\renewcommand*{\glossarymark}{}

\LWR@origpagestyle{empty}
\renewcommand*{\ps@empty}{}
\renewcommand*{\makepagestyle}[1]{}
\renewcommand*{\emptypshook}{}%
% \renewcommand*{\empty@oddhead}{}
% \renewcommand*{\empty@oddfoot}{}
% \renewcommand*{\empty@evenhead}{}
% \renewcommand*{\empty@evenfoot}{}
\renewcommand*{\@oddhead}{}
\renewcommand*{\@oddfoot}{}
\renewcommand*{\@evenhead}{}
\renewcommand*{\@evenfoot}{}
\renewcommand*{\aliaspagestyle}[2]{}
\renewcommand*{\copypagestyle}[2]{}

\renewcommand*{\makeevenhead}[4]{}
```

```
\renewcommand*{\makeoddhead}[4]{}
\renewcommand*{\makeevenfoot}[4]{}
\renewcommand*{\makeoddfoot}[4]{}
\renewcommand*{\makerunningwidth}[3]{}
% \newlength{\headwidth}
\renewcommand*{\makeheadrule}[3]{}
\renewcommand*{\makefootrule}[3]{}
\renewcommand*{\makeheadfootruleprefix}[3]{}
% \newlength{\normalrulethickness}
% \setlength{\normalrulethickness}{.4pt}
% \newlength{\footruleheight}
% \newlength{\footruleskip}
\renewcommand*{\makeheadposition}[5]{}
\renewcommand{\makepsmarks}[2]{}
\renewcommand*{\makeheadfootstrut}[3]{}

\renewcommand{\createplainmark}[3]{}
\renewcommand{\memUChead}[1]{}
\renewcommand{\createmark}[5]{}
\renewcommand*{\clearplainmark}[1]{}
\renewcommand*{\clearmark}[1]{}
\renewcommand{\addtopsmarks}[3]{}
\renewcommand{\ifonlyfloats}[2]{#2}
\renewcommand*{\mergepagefloatstyle}[3]{}

\renewcommand*{\framepichead}{}
\renewcommand*{\framepictextfoot}{}
\renewcommand*{\framepichook}{}
\renewcommand*{\showheadfootlocoff}{}
\renewcommand*{\showtextblocklocoff}{}
```

§ 422.9 Paragraphs and lists

```
\renewcommand{\hangfrom}[1]{#1}
\let\centerfloat\centering
\renewcommand*{\raggedyright}[1][]{}
% \newlength{\ragrparindent}
\renewcommand{\sourceatright}[2][]{\attribution{#2}}
\let\memorigdbs\LWR@endofline
\let\memorigpar\par
\let\atcentercr\LWR@endofline

\renewcommand*{\linenottooshort}[1][]{}
\renewcommand*{\russianpar}{}
\renewcommand*{\lastlinerulefill}{}
\renewcommand*{\lastlineparrule}{}
\renewcommand*{\justlastraggedleft}{}
\renewcommand*{\raggedrightthenleft}{}
\renewcommand*{\leftcenterright}{}
```

```
\renewcommand{\leftspringright}[4]{%
\begin{minipage}{#1\linewidth}#3\end{minipage}\qquad%
\begin{minipage}{#2\linewidth}\begin{flushright}#4\end{flushright}\end{minipage}%
}

\renewenvironment*{blockdescription}
{\LWR@descriptionstart\LWR@origdescription}
{\enddescription}
\renewcommand*{\blockdescriptionlabel}[1]{\textbf{#1}}
\renewenvironment*{labelled}[1]{\begin{description}}{\end{description}}
\renewenvironment*{flexlabelled}[6]{\begin{description}}{\end{description}}
\renewcommand*{\tightlists}{}
\renewcommand*{\defaultlists}{}
\RenewDocumentCommand{\firmlists}{s}{}
\renewcommand*{\firmlist}{}
\renewcommand*{\tightlist}{}
\renewcommand*{\zerotrivseps}{}
\renewcommand*{\savetrivseps}{}
\renewcommand*{\restoretrivseps}{}
```

§ 422.10 Contents lists

```
\csletcs{tableofcontents*}{tableofcontents}
\csletcs{listoffigures*}{listoffigures}
\csletcs{listoftables*}{listoftables}
\renewenvironment{KeepFromToc}{}{}
\renewcommand*{\onecoltocetc}{}
\renewcommand*{\twocoltocetc}{}
\renewcommand*{\ensureonecol}{}
\renewcommand*{\restorefromonecol}{}
\renewcommand*{\doccoltocetc}{}
\renewcommand*{\maxtocdepth}[1]{}% tocvsec2
\renewcommand*{\settocdepth}[1]{}% tocvsec2

\renewcommand{\tocheadstart}{}
\renewcommand{\printtoctitle}[1]{}
\renewcommand{\tocmark}{}
\renewcommand{\aftertoctitle}{}
\renewcommand{\lofheadstart}{}
\renewcommand{\printloftitle}[1]{}
\renewcommand{\lofmark}{}
\renewcommand{\afterloftitle}{}
\renewcommand{\lotheadstart}{}
\renewcommand{\printlottitle}[1]{}
\renewcommand{\lotmark}{}
\renewcommand{\afterlottitle}{}

\renewcommand*{\setpnumwidth}[1]{}
```

```
\renewcommand*{\setrmarg}[1]{}
\renewcommand*{\cftbookbreak}{}
\renewcommand*{\cftpartbreak}{}
\renewcommand*{\cftchapterbreak}{}

% \newlength{\cftbeforebookskip}
% \newlength{\cftbookindent}
% \newlength{\cftbooknumwidth}
\renewcommand*{\cftbookfont}{}
\renewcommand*{\cftbookname}{}
\renewcommand*{\cftbookpresnum}{}
\renewcommand*{\cftbookaftersnum}{}
\renewcommand*{\cftbookaftersnumb}{}
\renewcommand*{\cftbookleader}{}
\renewcommand*{\cftbookdotsep}{1}
\renewcommand*{\cftbookpagefont}{}
\renewcommand*{\cftbookafterpnum}{}
\renewcommand*{\cftbookformatpnum}[1]{}
\renewcommand*{\cftbookformatpnumhook}[1]{}
```

Part is already defined by **tocloft**.

```
% \newlength{\cftbeforechapterskip}
% \newlength{\cftchapterindent}
% \newlength{\cftchapternumwidth}
\renewcommand*{\cftchapterfont}{}
\renewcommand*{\cftchaptername}{}
\renewcommand*{\cftchapterpresnum}{}
\renewcommand*{\cftchapteraftersnum}{}
\renewcommand*{\cftchapteraftersnumb}{}
\renewcommand*{\cftchapterleader}{}
\renewcommand*{\cftchapterdotsep}{1}
\renewcommand*{\cftchapterpagefont}{}
\renewcommand*{\cftchapterafterpnum}{}
\renewcommand*{\cftchapterformatpnum}[1]{}
\renewcommand*{\cftchapterformatpnumhook}[1]{}

% \newlength{\cftbeforesectionskip}
% \newlength{\cftsectionindent}
% \newlength{\cftsectionnumwidth}
\renewcommand*{\cftsectionfont}{}
\renewcommand*{\cftsectionname}{}
\renewcommand*{\cftsectionpresnum}{}
\renewcommand*{\cftsectionaftersnum}{}
\renewcommand*{\cftsectionaftersnumb}{}
\renewcommand*{\cftsectionleader}{}
\renewcommand*{\cftsectiondotsep}{1}
\renewcommand*{\cftsectionpagefont}{}
\renewcommand*{\cftsectionafterpnum}{}
\renewcommand*{\cftsectionformatpnum}[1]{}
```

```
\renewcommand*{\cftsectionformatpnumhook}[1]{}

% \newlength{\cftbeforesubsectionskip}
% \newlength{\cftsubsectionindent}
% \newlength{\cftsubsectionnumwidth}
\renewcommand*{\cftsubsectionfont}{}
\renewcommand*{\cftsubsectionname}{}
\renewcommand*{\cftsubsectionpresnum}{}
\renewcommand*{\cftsubsectionaftersnum}{}
\renewcommand*{\cftsubsectionaftersnumb}{}
\renewcommand*{\cftsubsectionleader}{}
\renewcommand*{\cftsubsectiondotsep}{1}
\renewcommand*{\cftsubsectionpagefont}{}
\renewcommand*{\cftsubsectionafterpnum}{}
\renewcommand*{\cftsubsectionformatpnum}[1]{}
\renewcommand*{\cftsubsectionformatpnumhook}[1]{}

% \newlength{\cftbeforesubsubsectionskip}
% \newlength{\cftsubsubsectionindent}
% \newlength{\cftsubsubsectionnumwidth}
\renewcommand*{\cftsubsubsectionfont}{}
\renewcommand*{\cftsubsubsectionname}{}
\renewcommand*{\cftsubsubsectionpresnum}{}
\renewcommand*{\cftsubsubsectionaftersnum}{}
\renewcommand*{\cftsubsubsectionaftersnumb}{}
\renewcommand*{\cftsubsubsectionleader}{}
\renewcommand*{\cftsubsubsectiondotsep}{1}
\renewcommand*{\cftsubsubsectionpagefont}{}
\renewcommand*{\cftsubsubsectionafterpnum}{}
\renewcommand*{\cftsubsubsectionformatpnum}[1]{}
\renewcommand*{\cftsubsubsectionformatpnumhook}[1]{}

% \newlength{\cftbeforeparagraphskip}
% \newlength{\cftparagraphindent}
% \newlength{\cftparagraphnumwidth}
\renewcommand*{\cftparagraphfont}{}
\renewcommand*{\cftparagraphname}{}
\renewcommand*{\cftparagraphpresnum}{}
\renewcommand*{\cftparagraphaftersnum}{}
\renewcommand*{\cftparagraphaftersnumb}{}
\renewcommand*{\cftparagraphleader}{}
\renewcommand*{\cftparagraphdotsep}{1}
\renewcommand*{\cftparagraphpagefont}{}
\renewcommand*{\cftparagraphafterpnum}{}
\renewcommand*{\cftparagraphformatpnum}[1]{}
\renewcommand*{\cftparagraphformatpnumhook}[1]{}

% \newlength{\cftbeforesubparagraphskip}
% \newlength{\cftsubparagraphindent}
% \newlength{\cftsubparagraphnumwidth}
```

```
\renewcommand*{\cftsubparagraphfont}{}
\renewcommand*{\cftsubparagraphname}{}
\renewcommand*{\cftsubparagraphpresnum}{}
\renewcommand*{\cftsubparagraphaftersnum}{}
\renewcommand*{\cftsubparagraphaftersnumb}{}
\renewcommand*{\cftsubparagraphleader}{}
\renewcommand*{\cftsubparagraphdotsep}{1}
\renewcommand*{\cftsubparagraphpagefont}{}
\renewcommand*{\cftsubparagraphafterpnum}{}
\renewcommand*{\cftsubparagraphformatpnum}[1]{}
\renewcommand*{\cftsubparagraphformatpnumhook}[1]{}

% \newlength{\cftbeforefigureskip}
% \newlength{\cftfigureindent}
% \newlength{\cftfigurenumwidth}
\renewcommand*{\cftfigurefont}{}
\renewcommand*{\cftfigurename}{}
\renewcommand*{\cftfigurepresnum}{}
\renewcommand*{\cftfigureaftersnum}{}
\renewcommand*{\cftfigureaftersnumb}{}
\renewcommand*{\cftfigureleader}{}
\renewcommand*{\cftfiguredotsep}{1}
\renewcommand*{\cftfigurepagefont}{}
\renewcommand*{\cftfigureafterpnum}{}
\renewcommand*{\cftfigureformatpnum}[1]{}
\renewcommand*{\cftfigureformatpnumhook}[1]{}

% \newlength{\cftbeforesubfigureskip}
% \newlength{\cftsubfigureindent}
% \newlength{\cftsubfigurenumwidth}
\newcommand*{\cftsubfigurefont}{}
\newcommand*{\cftsubfigurename}{}
\newcommand*{\cftsubfigurepresnum}{}
\newcommand*{\cftsubfigureaftersnum}{}
\newcommand*{\cftsubfigureaftersnumb}{}
\newcommand*{\cftsubfigureleader}{}
\newcommand*{\cftsubfiguredotsep}{1}
\newcommand*{\cftsubfigurepagefont}{}
\newcommand*{\cftsubfigureafterpnum}{}
\newcommand*{\cftsubfigureformatpnum}[1]{}
\newcommand*{\cftsubfigureformatpnumhook}[1]{}

% \newlength{\cftbeforetableskip}
% \newlength{\cfttableindent}
% \newlength{\cfttablenumwidth}
\renewcommand*{\cfttablefont}{}
\renewcommand*{\cfttablename}{}
\renewcommand*{\cfttablepresnum}{}
\renewcommand*{\cfttableaftersnum}{}
\renewcommand*{\cfttableaftersnumb}{}
```

```
\renewcommand*{\cfttableleader}{}
\renewcommand*{\cfttabledotsep}{1}
\renewcommand*{\cfttablepagefont}{}
\renewcommand*{\cfttableafterpnum}{}
\renewcommand*{\cfttableformatpnum}[1]{}
\renewcommand*{\cfttableformatpnumhook}[1]{}

% \newlength{\cftbeforesubtableskip}
% \newlength{\cftsubtableindent}
% \newlength{\cftsubtablenumwidth}
\newcommand*{\cftsubtablefont}{}
\newcommand*{\cftsubtablename}{}
\newcommand*{\cftsubtablepresnum}{}
\newcommand*{\cftsubtableaftersnum}{}
\newcommand*{\cftsubtableaftersnumb}{}
\newcommand*{\cftsubtableleader}{}
\newcommand*{\cftsubtabledotsep}{1}
\newcommand*{\cftsubtablepagefont}{}
\newcommand*{\cftsubtableafterpnum}{}
\newcommand*{\cftsubtableformatpnum}[1]{}
\newcommand*{\cftsubtableformatpnumhook}[1]{}

\renewcommand*{\booknumberline}[1]{}
\renewcommand*{\partnumberline}[1]{}
\renewcommand*{\chapternumberline}[1]{}
\renewcommand*{\numberlinehook}[1]{}
% \renewcommand*{\cftwhatismyname}{}%
\renewcommand*{\booknumberlinehook}[1]{}
\renewcommand*{\partnumberlinehook}[1]{}
\renewcommand*{\chapternumberlinehook}[1]{}
\renewcommand{\numberlinebox}[2]{}
\renewcommand{\booknumberlinebox}[2]{}
\renewcommand{\partnumberlinebox}[2]{}
\renewcommand{\chapternumberlinebox}[2]{}
%
% \newlength{\cftparfillskip}
\renewcommand*{\cftpagenumbersoff}[1]{}
\renewcommand*{\cftpagenumberson}[1]{}
\renewcommand*{\cftlocalchange}[3]{}
\renewcommand*{\cftaddtitleline}[4]{}
\renewcommand*{\cftaddnumtitleline}[4]{}
\renewcommand{\cftinsertcode}[2]{}
\renewcommand{\cftinserthook}[2]{}
\renewcommand{\settocpreprocessor}[2]{}
\DeclareRobustCommand{\cftpagenumbersoff}[1]{}
\DeclareRobustCommand{\cftpagenumberson}[1]{}
```

§ 422.11 Floats and captions

`\newfloat` `[`⟨*1: within*⟩`]` `{`⟨*2: type*⟩`}` `{`⟨*3: ext*⟩`}` `{`⟨*4: capname*⟩`}`

```
\RenewDocumentCommand{\newfloat}{o m m m}{%
\IfValueTF{#1}%
{\DeclareFloatingEnvironment[fileext=#3,within=#1,name={#4}]{#2}}%
{\DeclareFloatingEnvironment[fileext=#3,name={#4}]{#2}}%
```

newfloat package automatically creates the `\listof` command for new floats, but **float** does not, so remove `\listof` here in case it is manually created later.

```
\cslet{listof#2s}\relax%
\cslet{listof#2es}\relax%
}
```

`\newlistof` `[`⟨*within*⟩`]` `{`⟨*type*⟩`}` `{`⟨*ext*⟩`}` `{`⟨*listofname*⟩`}`

Emulated through the `\newfloat` mechanism. Note that **memoir** uses a different syntax than **tocloft** for the name.

```
\RenewDocumentCommand{\newlistof}{o m m m}
{%
\IfValueTF{#1}
{\newlistentry[#1]{#2}{#3}{0}}
{\newlistentry{#2}{#3}{0}}
\@namedef{ext@#2}{#3}%
\@ifundefined{c@#3depth}{\newcounter{#3depth}}{}%
\setcounter{#3depth}{1}%
\@namedef{#3mark}{}%
\@namedef{#2}{\listof{#2}{#4}}
\@namedef{@cftmake#3title}{}
\@ifundefined{cftbefore#3titleskip}{
    \expandafter\newlength\csname cftbefore#3titleskip\endcsname
    \expandafter\newlength\csname cftafter#3titleskip\endcsname
}{}
\@namedef{cft#3titlefont}{}
\@namedef{cftafter#3title}{}
\@namedef{cft#3prehook}{}
\@namedef{cft#3posthook}{}
}

\renewcommand{\setfloatadjustment}[2]{}
```

Borrowed from the **lwarp** version of **keyfloat**:

```
\NewDocumentEnvironment{KFLTmemoir@marginfloat}{O{-1.2ex} m}
{% start
\LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
\captionsetup{type=#2}%
}
{%
\endLWR@BlockClassWP%
}
```

```
\DeclareDocumentEnvironment{marginfigure}{o}
  {\begin{KFLTmemoir@marginfloat}{figure}}
  {\end{KFLTmemoir@marginfloat}}

\DeclareDocumentEnvironment{margintable}{o}
  {\begin{KFLTmemoir@marginfloat}{table}}
  {\end{KFLTmemoir@marginfloat}}

\renewcommand{\setmarginfloatcaptionadjustment}[2]{}
\renewcommand{\setmpjustification}[2]{}
\renewcommand*{\mpjustification}{}
\renewcommand*{\setfloatlocations}[2]{}
\DeclareDocumentCommand{\suppressfloats}{o}{}
\renewcommand*{\FloatBlock}{}
\renewcommand*{\FloatBlockAllowAbove}{}
\renewcommand*{\FloatBlockAllowBelow}{}
\renewcommand*{\setFloatBlockFor}{}

\renewcommand{\captiontitlefinal}[1]{}

\renewcommand{\flegtable}{\tablename}
\renewcommand{\flegfigure}{\figurename}
\renewcommand{\flegtoctable}{}
\renewcommand{\flegtocfigure}{}


\renewcommand{\subcaption}[2][]{%
\ifblank{#1}{\subfloat[#2]{}}{\subfloat[#1][#2]{}}%
}

\renewcommand{\contsubcaption}{\ContinuedFloat\subcaption}

\LetLtxMacro\subcaptionref\subref

\renewcommand*{\tightsubcaptions}{}
\renewcommand*{\loosesubcaptions}{}

\renewcommand*{\subcaptionsize}[1]{}
\renewcommand*{\subcaptionlabelfont}[1]{}
\renewcommand*{\subcaptionfont}[1]{}
\renewcommand*{\subcaptionstyle}[1]{}

\renewcommand*{\hangsubcaption}{}
\renewcommand*{\shortsubcaption}{}
\renewcommand*{\normalsubcaption}{}

\RenewDocumentEnvironment{sidecaption}{o m o}
{}
```

```
{
\IfValueTF{#1}{\caption[#1]{#2}}{\caption{#2}}%
\IfValueT{#3}{\label{#3}}%
}

% \newlength{\sidecapwidth}
% \newlength{\sidecapsep}
\renewcommand*{\setsidecaps}[2]{}
\renewcommand*{\sidecapmargin}[1]{}
% \newif\ifscapmargleft
\scapmargleftfalse
\renewcommand*{\setsidecappos}[1]{}

\RenewDocumentEnvironment{sidecontcaption}{m o}
{}
{%
\ContinuedFloat%
\caption{#1}%
```

Without \@captype, the section is referred to instead.

```
\IfValueT{#2}{\label[\@captype]{#2}}%
}
```

\sidenamedlegend does not appear to use the TOC argument.

```
\renewenvironment{sidenamedlegend}[2][]{
\begin{center}
\@nameuse{\@captype name}\CaptionSeparator#2
\end{center}
}
{}

\renewenvironment{sidelegend}[1]
{\begin{center}
#1

}
{\end{center}}

\renewcommand*{\sidecapstyle}{}
\renewcommand*{\overridescapmargin}[1]{}
% \newlength{\sidecapraise}
\renewcommand*{\sidecapfloatwidth}{\linewidth}

\LetLtxMacro\ctabular\tabular
\LetLtxMacro\endctabular\endtabular

\renewcommand{\autorows}[5][]{%
```

```
#5
}

\renewcommand{\autocols}[5][]{%
#5
}
```

§ 422.12 Page notes

```
\renewcommand*{\feetabovefloat}{}
\renewcommand*{\feetbelowfloat}{}
\renewcommand*{\feetatbottom}{}

\renewcommand*{\verbfootnote}[2][]{
\PackageError{lwarp,memoir}
{Verbatim footnotes are not yet supported by lwarp.}
{This may be improved some day.}
}

\renewcommand*{\plainfootnotes}{}
\renewcommand*{\twocolumnfootnotes}{}
\renewcommand*{\threecolumnfootnotes}{}
\renewcommand*{\paragraphfootnotes}{}
\renewcommand*{\footfudgefiddle}{}

\renewcommand*{\newfootnoteseries}[1]{
\PackageError{lwarp,memoir}
{Memoir footnote series are not yet supported by lwarp.}
{This may be improved some day.}
}

\renewcommand*{\plainfootstyle}[1]{}
\renewcommand*{\twocolumnfootstyle}[1]{}
\renewcommand*{\threecolumnfootstyle}[1]{}
\renewcommand*{\paragraphfootstyle}[1]{}

\renewcommand*{\footfootmark}{}
\renewcommand*{\footmarkstyle}[1]{}

% \newlength{\footmarkwidth}
% \newlength{\footmarksep}
% \newlength{\footparindent}

\renewcommand*{\foottextfont}{}

\renewcommand*{\marginparmargin}[1]{}
\renewcommand*{\sideparmargin}[1]{}

\LetLtxMacro\sidepar\marginpar
```

```
\renewcommand*{\sideparfont}{}
\renewcommand*{\sideparform}{}
\LWR@providelength{\sideparvshift}

\renewcommand*{\parnopar}{}

\renewcommand{\sidebar}[1]{\begin{quote}#1\end{quote}}
\renewcommand*{\sidebarmargin}[1]{}
\renewcommand*{\sidebarfont}{}
\renewcommand*{\sidebarform}{}
% \newlength{\sidebarhsep}
% \newlength{\sidebarvsep}
% \newlength{\sidebarwidth}
% \newlength{\sidebartopsep}
\renewcommand{\setsidebarheight}[1]{}
\renewcommand*{\setsidebars}[6]{}
\renewcommand*{\footnotesatfoot}{}
\renewcommand*{\footnotesinmargin}{}

\LetLtxMacro\sidefootnote\footnote
\LetLtxMacro\sidefootnotemark\footnotemark
\LetLtxMacro\sidefootnotetext\footnotetext

\renewcommand*{\sidefootmargin}[1]{}
% \newlength{\sidefoothsep}
% \newlength{\sidefootvsep}
% \newlength{\sidefootwidth}
% \newlength{\sidefootadjust}
% \newlength{\sidefootheight}
\renewcommand*{\setsidefootheight}[1]{}
% \renewcommand*{\sidefootfont}{}% in docs but not in the package
\renewcommand*{\setsidefeet}[6]{}
\renewcommand*{\sidefootmarkstyle}[1]{}
\renewcommand*{\sidefoottextfont}{}
\renewcommand*{\sidefootform}{}

\renewcommand*{\continuousnotenums}{\pncontopttrue}% from pagenote
\renewcommand*{\notepageref}{}
\renewcommand*{\prenotetext}{}
\renewcommand*{\postnotetext}{}
\renewcommand*{\idtextinnotes}[1]{}
\renewcommand*{\printpageinnotes}[1]{}
\renewcommand*{\printpageinnoteshyperref}[1]{}
\renewcommand*{\foottopagenote}{}
\renewcommand*{\pagetofootnote}{}
```

§ 422.13 Decorative text

```
\renewcommand*{\epigraphposition}[1]{}
```

```
\renewcommand*{\epigraphtextposition}[1]{}
\renewcommand*{\epigraphsourceposition}[1]{}
\renewcommand*{\epigraphfontsize}[1]{}
\renewcommand*{\epigraphforheader}[2][]{}
\renewcommand*{\epigraphpicture}{}
```

§ 422.14 Poetry

```
\renewcommand*{\vinphantom}{}
\renewcommand*{\vleftofline}[1]{#1}
% \let\linenumberfrequency\poemlines
% \renewcommand*{\linenumberfont}[1]{}

\DeclareDocumentCommand{\PoemTitle}{s o o m}{%
\IfValueTF{#2}%
{\poemtitle[#2]{#4}}%
{\poemtitle{#4}}%
}

\renewcommand*{\NumberPoemTitle}{}
\renewcommand*{\PlainPoemTitle}{}
\renewcommand*{\poemtitlepstyle}{}
\renewcommand*{\poemtitlestarmark}[1]{}
\renewcommand*{\poemtitlestarpstyle}{}
\renewcommand*{\PoemTitleheadstart}{}
\renewcommand*{\printPoemTitlenonum}{}
\renewcommand*{\printPoemTitlenum}{}
\renewcommand*{\afterPoemTitlenum}{}
\renewcommand*{\printPoemTitletitle}[1]{}
\renewcommand*{\afterPoemTitle}{}
\newlength{\midpoemtitleskip}
\renewcommand*{\PoemTitlenumfont}{}
\renewcommand*{\PoemTitlefont}{}
```

§ 422.15 Boxes, verbatims and files

```
\renewenvironment{qframe}{\framed}{\endframed}
\renewenvironment{qshade}{\shaded}{\endshaded}
```

Use the **comment** package:

```
\renewcommand*{\commentsoff}[1]{\includecomment{#1}}
\renewcommand*{\commentson}[1]{\excludecomment{#1}}
\LetLtxMacro\renewcomment\commentson

\renewcommand*{\setverbatimfont}[1]{}
\renewcommand*{\tabson}[1]{}
\renewcommand*{\tabsoff}{}
\renewcommand*{\wrappingon}{}
```

```
\renewcommand*{\wrappingoff}{}
\renewcommand*{\verbatimindent}{}
\renewcommand*{\verbatimbreakchar}[1]{}

\DefineVerbatimEnvironment{fboxverbatim}{Verbatim}{frame=single}
```

`boxedverbatim` is already defined by **moreverb**. `boxedverbatim*` does not appear to work at all, even in a minimal print **memoir** document.

```
\renewcommand*{\bvbox}{}
\renewcommand*{\bvtopandtail}{}
\renewcommand*{\bvsides}{}
\renewcommand*{\nobvbox}{}
% \newlength\bvboxsep
\renewcommand*{\bvtoprulehook}{}
\renewcommand*{\bvtopmidhook}{}
\renewcommand*{\bvendrulehook}{}
\renewcommand*{\bvleftsidehook}{}
\renewcommand*{\bvrightsidehook}{}
\renewcommand*{\bvperpagetrue}{}
\renewcommand*{\bvperpagefalse}{}
\renewcommand{\bvtopofpage}[1]{}
\renewcommand{\bvendofpage}[1]{}
\renewcommand*{\linenumberfrequency}[1]{}
\renewcommand*{\resetbvlinenumber}{}
\renewcommand*{\setbvlinenums}[2]{}
\renewcommand*{\linenumberfont}[1]{}
\renewcommand*{\bvnumbersinside}{}
\renewcommand*{\bvnumbersoutside}{}
```

§ 422.16 Cross referencing

```
\renewcommand*{\fref}[1]{\cref{#1}}
\renewcommand*{\tref}[1]{\cref{#1}}
\renewcommand*{\pref}[1]{\cpageref{#1}}
\renewcommand*{\Aref}[1]{\cref{#1}}
\renewcommand*{\Bref}[1]{\cref{#1}}
\renewcommand*{\Pref}[1]{\cref{#1}}
\renewcommand*{\Sref}[1]{\cref{#1}}
\renewcommand*{\figurerefname}{Figure}
\renewcommand*{\tablerefname}{Table}
\renewcommand*{\pagerefname}{page}
\renewcommand*{\bookrefname}{Book~}
\renewcommand*{\partrefname}{Part~}
\renewcommand*{\chapterrefname}{Chapter~}
\renewcommand*{\sectionrefname}{\S}
\renewcommand*{\appendixrefname}{Appendix~}
\LetLtxMacro\titleref\nameref
\renewcommand*{\headnameref}{}
```

```
\renewcommand*{\tocnameref}{}

\providecounter{LWR@currenttitle}

\renewcommand*{\currenttitle}{%
    \addtocounter{LWR@currenttitle}{1}%
    \label{currenttitle\arabic{LWR@currenttitle}}%
    \nameref{currenttitle\arabic{LWR@currenttitle}}%
}

\renewcommand*{\theTitleReference}[2]{}
\renewcommand*{\namerefon}{}
\renewcommand*{\namerefoff}{}
```

§ 422.17 Back matter

Redefined to write the LWR@autoindex counter instead of page. Note that **memoir** has two versions, depending on the use of **hyperref**.

```
\AtBeginDocument{

\def\@@wrindexhyp#1||\\{%
    \addtocounter{LWR@autoindex}{1}%
    \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
%     \ifshowindexmark\@showidx{#1}\fi
    \protected@write\@auxout{}%
    {\string\@@wrindexm@m{\@idxfile}{#1}{\arabic{LWR@autoindex}}}%
    \endgroup
    \@esphack}%
```

\specialindex behaves like a regular \index, pointing to where \specialindex is used. If \specialindex is used inside a figure or table after the \caption, then the hyperlink will be given the name of that particular figure or table.

```
\def\@@wrspindexhyp#1||\\{%
    \addtocounter{LWR@autoindex}{1}%
    \LWR@new@label{LWRindex-\arabic{LWR@autoindex}}%
%     \ifshowindexmark\@showidx{#1}\fi
    \protected@write\@auxout{}%
    {%
%         \string\@@wrindexm@m{\@idxfile}{#1}{\@nameuse{the\@sptheidx}}%
        \string\@@wrindexm@m{\@idxfile}{#1}{\arabic{LWR@autoindex}}%
    }%
    \endgroup
    \@esphack}%

}% \AtBeginDocument
```

Patched to use _html filename and \BaseJobname:

```
\catcode`\_=12%
\renewcommand*{\makeindex}[1][\BaseJobname]{%
  \if@filesw
    \def\index{\@bsphack%
      \@ifnextchar [{\@index}{\@index[\BaseJobname]}}
    \def\specialindex{\@bsphack\@spindex}%
    \makememindexhook
    \expandafter\newwrite\csname #1@idxfile\endcsname
    \expandafter\immediate\openout \csname #1@idxfile\endcsname #1_html.idx\relax
    \typeout{Writing index file #1_html.idx }%
  \fi}
\catcode`\_=8%
```

Patched to use _html filename and \BaseJobname. This will later be patched by the **lwarp** core.

```
\catcode`\_=12%
\renewcommand{\printindex}[1][\BaseJobname]{\@input@{#1_html.ind}}
\catcode`\_=8%

\DeclareDocumentCommand{\newblock}{}{}
%
\renewcommand*{\showindexmarks}{}
\renewcommand*{\hideindexmarks}{}

\renewcommand*{\xindyindex}{}
```

§ 422.18 Miscellaneous

```
\renewcommand*{\changemarks}{}
\renewcommand*{\nochangemarks}{}
\renewcommand*{\added}[1]{}
\renewcommand*{\deleted}[1]{}
\renewcommand*{\changed}[1]{}

\renewcommand*{\showtrimsoff}{}
\renewcommand*{\showtrimson}{}
\renewcommand*{\trimXmarks}{}
\renewcommand*{\trimLmarks}{}
\renewcommand*{\trimFrame}{}
\renewcommand*{\trimNone}{}
\renewcommand*\trimmarkscolor{}
\renewcommand*{\trimmarks}{}
\renewcommand*{\tmarktl}{}
\renewcommand*{\tmarktr}{}
\renewcommand*{\tmarkbr}{}
\renewcommand*{\tmarkbl}{}
```

```
\renewcommand*{\tmarktm}{}
\renewcommand*{\tmarkmr}{}
\renewcommand*{\tmarkbm}{}
\renewcommand*{\tmarkml}{}
\renewcommand*{\trimmark}{}
\renewcommand*{\quarkmarks}{}
\renewcommand*{\registrationColour}[1]{}

\renewcommand*{\leavespergathering}[1]{}

\renewcommand*{\noprelistbreak}{}

\renewcommand*{\cleartorecto}{}
\renewcommand*{\cleartoverso}{}

\renewenvironment{vplace}[1][]{}{}
```

§ 422.19 Ccaption emulation

```
\renewcommand*{\captiondelim}[1]{\renewcommand*{\CaptionSeparator}{#1}}
\renewcommand*{\captionnamefont}[1]{}
\renewcommand*{\captiontitlefont}[1]{}
\renewcommand*{\flushleftright}{}
\renewcommand*{\centerlastline}{}
\renewcommand*{\captionstyle}[2][]{}
\DeclareDocumentCommand{\captionwidth}{m}{}
\renewcommand*{\changecaptionwidth}{}
\renewcommand*{\normalcaptionwidth}{}
\renewcommand*{\hangcaption}{}
\renewcommand*{\indentcaption}[1]{}
\renewcommand*{\normalcaption}{}
\renewcommand{\precaption}[1]{}
\renewcommand{\postcaption}[1]{}
\renewcommand{\midbicaption}[1]{}
\renewcommand{\contcaption}[1]{%
% \ContinuedFloat%
% \caption{#1}%
\begin{LWR@figcaption}% later becomes \caption*
\@nameuse{\@captype name} \thechapter.\the\value{\@captype}\CaptionSeparator #1
\end{LWR@figcaption}
}

\newlength{\abovelegendskip}
\setlength{\abovelegendskip}{0.5\baselineskip}
\newlength{\belowlegendskip}
\setlength{\belowlegendskip}{\abovelegendskip}
```

The extra \\ here forces a <br> in HTML when \legend is used in a \marginpar.

```
\renewcommand{\legend}[1]{\begin{center}#1\\\end{center}}

\renewcommand{\namedlegend}[2][]{
\begin{center}
\@nameuse{fleg\@captype}\CaptionSeparator#2\\
\end{center}
\@nameuse{flegtoc\@captype}{#1}
}

\renewcommand{\newfixedcaption}[3][\caption]{%
  \renewcommand{#2}{\def\@captype{#3}#1}}
\renewcommand{\renewfixedcaption}[3][\caption]{%
  \renewcommand{#2}{\def\@captype{#3}#1}}
\renewcommand{\providefixedcaption}[3][\caption]{%
  \providecommand{#2}{\def\@captype{#3}#1}}

\renewcommand{\bitwonumcaption}[6][]{%
\ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
\addtocounter{\@captype}{-1}%
\begingroup%
\csdef{\@captype name}{#4}%
\ifblank{#5}{\caption{#6}}{\caption[#5]{#6}}%
\endgroup%
\ifblank{#1}{}{\label{#1}}%
}

\LetLtxMacro\bionenumcaption\bitwonumcaption% todo

\renewcommand{\bicaption}[5][]{%
\ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
\begin{LWR@figcaption}% later becomes \caption*
#4 \thechapter.\the\value{\@captype}\CaptionSeparator #5
\end{LWR@figcaption}
\ifblank{#1}{}{\label{#1}}%
}

\renewcommand{\bicontcaption}[3]{%
\contcaption{#1}%
\begingroup%
\csdef{\@captype name}{#2}%
\contcaption{#3}%
\endgroup%
}

% only in ccaption, not in memoir:
% \LetLtxMacro\longbitwonumcaption\bitwonumcaption%
% \LetLtxMacro\longbionenumcaption\bitwonumcaption%
% \LetLtxMacro\longbicaption\bicaption%

\RenewDocumentCommand{\subtop}{O{} O{} m}{%
```

```
\subfloat[#1][#2]{#3}%
}

\RenewDocumentCommand{\subbottom}{O{} O{} m}{%
\subfloat[#1][#2]{#3}%
}

\renewcommand{\contsubtop}{%
\ContinuedFloat\addtocounter{\@captype}{1}%
\subtop}

\renewcommand{\contsubbottom}{%
\ContinuedFloat\addtocounter{\@captype}{1}%
\subbottom}

\renewcommand{\subconcluded}{}

\let\subfigure\subbottom
\let\subtable\subtop

\let\contsubtable\contsubtop
\let\contsubfigure\contsubbottom

\newcommand{\newfloatentry}[4][\@empty]{TODO: newfloatentry}
\newcommand{\newfloatlist}[5][\@empty]{TODO: newfloatlist}
\newcommand{\newfloatenv}[4][\@empty]{TODO: newfloatenv}
\DeclareRobustCommand{\newfloatpagesoff}[1]{}
\DeclareRobustCommand{\newfloatpageson}[1]{}
\newcommand{\setnewfloatindents}[3]{}
```

§ 422.20 Final patchwork

```
\newlistof{tableofcontents}{toc}{\contentsname}
\newlistof{listoffigures}{lof}{\listfigurename}
\newlistof{listoftables}{lot}{\listtablename}
```

Change History and Index

For the most recent changes, followed by the start of the Index, see page 997.

§ 422 Change History

v0.10
General: 2016/03/08 Initial version . 1
v0.11
General: 2016/03/11 1
Added section: Operating-System portability. 187
Added section: Selecting the operating system. 113
Test Suite: MS-WINDOWS in README.txt 1
Test Suite: limages and index in README.txt 1
v0.12
\LWR@newhtmlfile: Bugfix: TOC with numbered files. 342
General: 2016/03/14 1
Global: Uses \p@(type) in float captions. 1
Test Suite: Sub-figures 1
v0.13
\CaptionSeparator: Fix for newer babel package. 470
\LWR@LwarpStart: \up and \fup . . 359
General: 2016/03/24 1
Fix dollar-redefined bug for newer package. 879
Removed package: subfig 1
Test Suite: Ordinals, Subcaption . . 1
v0.14
\LWR@htmlsectionfilename: Fix: Links to home page. 306
General: 2016/03/31 1
floatrow: Added. 696
Docs: Commands for a successful HTML conversion. 117
Docs: Commands into a warpprint environment. 115
Docs: Newclude limitations. . . . 161
Docs: Table: Cross-referencing data structures. 454
Docs: Table: Float data structures. 466
Docs: Trademarks section. 572
Docs: Troubleshooting cross-references. 176
Test Suite: Assigned cleveref name for Test Float. 1
Test Suite: Floatrow 1
v0.15
General: 2016/04/06 1
Added. 701
Ampersand (&): Fixed handling when passed as an argument. . 397
Docs: Added warning icons for items needing special attention. 183
Docs: Clarify print/HTML output. 114
Docs: Moved the supported features table to the introduction. 69
Files: lwarp_formal.css added. . . . 1
Fix: steps counter 701
Fixed & handling. 699
Test Suite: test_suite_formal.css file added. 1
v0.16
General: 2016/04/11 1
\titlingpage: Improved print-output spacing. 368
xfrac: Adjusted for the use of any font: 935
Added XeLaTeX, LuaLaTeX support. 184
Docs: Font and UTF-8 support. . 100
Docs: Moved location of \usepackage{lwarp}. 102
Docs: Text not converting. 176
Lwarp no longer selects fonts. 100, 204
Removed package: suffix 1

Test Suite: Improved titlingpage. 368
Test Suite: Lwarp no longer selects fonts. 1
Test Suite: Supports XeLaTeX, LuaLaTeX. 1
v0.17
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page. 306
General: 2016/04/14 1
mdframed: Added. 771
Test Suite: Fix: Print-version front-matter page numbers. 1
Test Suite: Mdframed 1
v0.18
\LWR@includegraphicsb: Add: svgz file extension. 721
em, ex, %, px dimensions preserved. 721
Fix: \linewidth, \textwidth, \textheight inside a minipage. 721
Improved HTML output linebreaks. 721
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic. . . . 475
\LWR@newhtmlfile: sideTOC after title, improving responsive design. 341
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic. . . . 361
\LWR@subhyperref: Improved HTML output linebreaks. 462
\LWR@subhyperrefclass: Improved HTML output linebreaks. 463
\LWR@subinlineimage: Surpress extra space. 464
\hspace: \hspace supported. 562
General: 2016/05/19 1
File: lwarp.css: Improved TOC outline display. 1
Files: lwarp.css and lwarp_formal.css: Improved responsive design. 1
Microtype disabled during HTML generation 204
PDF Unicode input characters. . 185
Test Suite: Verse package 1
lateximage: pdfcrop: --hires added. 522
Reorganize \HomeHTMLFilename logic. 522
Surpress extra space. 522
verse: Supports verse, memoir packages. 914
minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage. 543
v0.19
\HTMLFilename: Docs: Escape filename underscores. 305
\HomeHTMLFilename: Docs: Escape filename underscores. 305
\LWR@LwarpStart: Enabled \\ equal to \newline. 358
\LWR@doequation: MathJax support. 508
\LWR@doubledollar: MathJax support. 501
\LWR@filestart: lwarp_mathjax.txt loaded. 356
\LWR@minipagestartpars: Surpresses paragraph tags between minipages. 561
\LWR@subsingledollar: MathJax support. 495
\LateximageFontSizeName: Add: User-adjustable math/lateximage font size. 518
\hspace: Fix: \hspace length computations. 562
\minipagefullwidth: Added: No width tag for the next minipage in HTML. 543
\warpHTMLonly: Added. 194
\warpprintonly: Replaces \rowprintedonly. 194
\xfracHTMLfontsize: Added. . . . 934
General: 2016/06/08 1
MathJax support added. 505, 512, 513
multirow: Added optional args. . 791
Adapts to tikz version. 879
Avoids MathJax. 494
cleveref: Loaded \AtEndPreamble. 538
CSS for table note item. 878
Docs: Math options. 102
Docs: Table: Cross-referencing data structures, updated. . . . 454
File: lwarp.css: tnoteitemheader added. 1
File: lwarp_mathjax.txt added. . 1

Introduction: MathJax support mentioned. 67
Options: mathsvg and mathjax . 189
Supports colored \rule. 926
titleps: null \pagestyle and \thispagestyle for HTML. . . 880
v0.20
\BlockClassSingle: Renamed from "LWR@htmldivclassline". 318
\HTMLDescription: Added \NewHTMLdescription. (Renamed in v0.30.) 329
\HTMLFilename: No longer escape underscores. 305
\HomeHTMLFilename: No longer escape underscores. 305
\InlineClass: Renamed from "inlineclass". 318
\LWR@LwarpStart: Fix: math cross references. 360
\LWR@closeparagraph: \unskip extra spaces. 322
No break tags in the start/end of a tabular. 322
\LWR@endofline: Fix: \\ 561
\LWR@filestart: Adds meta description. 356
\LWR@htmldivclass: Added optional style. 316
\LWR@htmlelementclass: Added optional style. 316
\LWR@htmlsectionfilename: HTMLFilename: removed additional trailing '-', and may be empty. 306
Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix. . . . 306
\LWR@includegraphicsb: Fix: \linewidth in a floatrow. 721
Fix: Expands filename. 721
\LWR@longtabledatacaptiontag: Fix: Pars in captions. 437
\LWR@section: Combined higher-level sections together into files. 348
\LWR@setOSWindows: Auto-detects operating system. 188
\LWR@subhtmlelementclass: Factored code. 316
\SetHTMLFileNumber: Add: Control file numbers. 305
\cpagerefFor: User-redefinable word for page references. . . . 539
\dotfill: Inserts an ellipsis. 560
\hfill: Inserts a \qquad. 560
\hrulefill: Inserts a short rule. . . 560
\hspace: Add: Supports HTML thin breakable space. 562
\hyperindexref: Print mode provided in case **hyperref** not used. 485
\pageref: Added. 461
\tracinglwarp: Added. 214
General: 2017/02/09 1
afterpage: Added. 582
alltt: Added. 587
bookmark: Added. 613
caption and **subcaption** supported. 1
cleveref and referencing patches: Applied \AfterEndPreamble. . 538
draftwatermark: Added. 663
eso-pic: Added. 673
everypage: Added. 674
extramarks: Added. 674
fancyhdr: Added. 681
float: Improved float caption type handling. 693
hyperref: Additional user macros. 734
keyfloat: Added. 750
letterspace: User-interface emulated. 753
listings: Added. 757
ltcaption: Added. 765
lwarp-newproject: Added. 227
microtype: User-interface emulated. 785
needspace: Added. 796
nowidow: Added. 799
placeins: Added. 822
ragged2e: Added. 825
setspace: Improved support. . . . 842
textpos: Added. 872
titleps: Added. 880
titlesec: Added. 884
titletoc: Added. 886
titling: Improved compatibility. . 888
tocloft: Added. 896
wallpaper: Added. 919

wrapfig: Added. 921
Added @, <, > columns. 391
Added single-expansion data arrays. 300
Code factored into independent `lwarp_html` files. 572
Docs: Examples for generating HTML file names. 111
Docs: Improved index. 1
Enhanced **titling** support. 366
File: lwarp.css: Minor fixes for validation. 1
File: lwarpmk used to compile print, HTML, indexes, and lateximages. 1
Fix: `\linewidth` in a floatrow. . . 699
Moved sidebar and example code to test suite. 1
Page geometry set to 6in wide with large margins. 205
Parallel versions of aux files for print/HTML. 1
Removed reliance on make, grep, gawk. 1
Tabular: `\unskip` extra spaces. . . 391
Test Suite: HTML meta descriptions. 1
`verbatim`: Added. 378
`BlockClass`: Added optional style. . 317
Renamed from "blockclass". . . . 317
`LWR@nestspan`: Fix: Minipages inside a span. 313

v0.21

`\LWR@LwarpStart`: Changed `lateximages` to a `.txt` file. . . 358
`\LWR@filestart`: Skip title if not given. 356
`\LWR@newhtmlfile`: Skip title if not given. 341
`\marginpar`: Fixed source listing. . . 335
`\marginparBlock`: Fixed source listing. 335
General: 2017/02/23 1
fontenc: Added. 704
fontspec: Added. 705
inputenc: Added. 749
newclude: Added. 796
newunicodechar: Added. 796
lwarpmk: Fix: *`lwarpmk again`* for WINDOWS. 279
lwarpmk: Fix: *`lwarpmk limages`* for WINDOWS. 279
lwarpmk: Fix: **lwarpmk** uses `lateximages` text file instead of shell script. 279
Add: Errors for misplaced packages. 195
Docs: Added **internet** class. . . . 74
Docs: Added TeX2page, GladTeX. 74
Docs: Installing on WINDOWS. . . 81
File: `lwarp_tutorial.txt` added. 85

v0.22

`\LWR@parseDcolumn`: Added tabular `D` column. 407
`\LWR@parsebangcolumn`: Added tabular `!` column. 402
`\LWR@parsetablecols`: Unknown table column types become `l`. Added tabular `D`, `!`, `X` columns. 409
`\LWR@printmccoldata`: Added tabular `D`, `!`, and `X` columns. . . 431
General: 2017/03/02 1
abstract: Added. 574
changepage: Added. 625
dcolumn: Added. 660
ftnright: Added. 711
geometry: Nullified commands. . 713
indentfirst: Added. 747
layout: Added. 752
lscape: Added. 764
mcaption: Added. 771
nameref: Added. 794
nextpage: Added. 797
parskip: Added. 815
showkeys: Added. 844
sidecap: Added. 845
tabularx: Added. 867
varioref: Supported. 125
verse: Added. 913

v0.23

`\LWR@parsetablecols`: Fix for vert bar column type. 409
`\LWR@printmccoldata`: Fix for vert bar column type. 431
General: 2017/03/02 1

v0.24

`\LWR@htmlfileref`: Fix: Index links while `\tracinglwarp`. 457
`\hspace`: Add: `\hspace \fill` converts to 2em 562

\hypertocfloat: List of floats responds to lofdepth, lotdepth. 481
General: 2017/03/15 1
floatrow: Support for **subfig**. . . . 696
subfig: Added. 859
tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary. 879
Docs: Filename underscore. 106, 120
Fix for inline images. 879
No longer preloads **subcaption**; conflicted with **subfig**. 207
picture: Fix for inline images. . . . 541
v0.25
\LWR@loadnever: Added the ability to prevent conflicting packages. 196
\addcontentsline: Handles theorems. 473
General: 2016/03/22 1
amsthm: Added. 594
ccaption: Prevented. 624
ellipsis: Added. 665
emptypage: Added. 665
framed: Added. 709
lips: Added. 757
mdframed: Help avoid hyphenation. 773
ntheorem: Added. 799
showidx: Added. 844
theorem: Added. 873
Basic LaTeX theorems: improved css. 379
Docs: Adds credits for patched code. 1
Docs: Testing **lwarp**. 174
Fix: Allows XeLaTeX and LuaLaTeX to preload graphics and graphicx. 197
v0.26
General: 2017/03/31 1
lwarp.css: Improved responsive marginpar and marginblock. . 235
cutwin: Added. 659
endnotes: Added. 667
floatflt: Added. 695
footmisc: Added. 705
footnotehyper: Added. 707
footnote: Added. 706
marginfix: Added. 770
marginnote: Added. 770
mparhack: Added. 788
pagenote: Supported as-is. . . . 813
sidenotes: Added. 846
Docs: Improved MiKTeX install instructions. 79, 81
Dollar span avoided in a lateximage. 494
Footnotes now are LaTeX boxes instead of pagenotes. 330
lateximage: Labels track page numbers of lateximages. 522
Print mode now uses a minipage of \linewidth. 522
picture: Fix for \makebox in picture. 541
v0.27
\LWR@footnotetext: Fix for table footnote par tags. 331
General: 2017/04/04 1
lettrine: Added. 753
microtype: Fix with XeLaTeX, LuaLaTeX. 785
soul: Added. 853
ulem: Added. 909
Docs: Installing utilities for MacOS. 83
Docs: Limitations of saveboxes. . 120
Page geometry modified to reduce line overflow. 205
v0.28
\@wrindex: Improved indexing. . . . 484
\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.) 329
\LWR@LwarpEnd: If FormatEPUB or FormatWP, no bottom nav. . . . 362
\LWR@LwarpStart: FormatWordProcessor forces single-file output. 358
\LWR@filestart: Adds HTML meta author. 356
\LWR@forcenewpage: Forces new PDF page before major environments. 309
\LWR@htmlcomment: Breaks ligatures in HTML comments. 315
\LWR@includegraphicsb: Adapts to **graphics** syntax. 721
\LWR@newhtmlfile: If FormatEPUB or FormatWP: skips headers, footers, nav. 341

\LWR@parsetablecols: Added L, C, R, J column types. 409
\LWR@startref: Removed space. . . 459
\chapter: If EPUB, prints footnotes before each section. 353
\hyperindexref: Improved indexing. 485
\textup: Fixed span class. 553
General: 2017/04/14 1
glossaries: Added. 714
graphics: Added. 716
tabularx: Fix for optional pos. . . 867
tabulary: Added. 868
lwarpmk: Add: *printglossary* and *htmlglossary* commands. 279
Added boolean FormatEPUB. . . . 220
Added boolean FormatWP. 220
Added boolean HTMLDebugComments. 214
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42. 220
Docs: Modfying lwarpmk and index processing. 175
File: lwarp_mathjax.txt: Updated CDN repository. 276
Forced oneside to maintain large right margin. 205
v0.29
\LWR@includegraphicsb: Fix: Error when no optional arguments. . 721
General: 2017/04/15 1
*.lwarpmkconf: Add: language option for config files. 234
lwarpmk.conf: Add: language option for config files. 234
lwarpmk: Add: language option for config files. 279
Add: lwarpmklang option for **lwarp**. 190
Docs: Using a glossary 96
v0.30
\CSSFilename: Renamed from \NewCSS. 328
\HTMLAuthor: Renamed from \HTMLauthor. 329
\HTMLDescription: Renamed from \NewHTMLdescription. 329
\HTMLFirstPageTop: Renamed from \SetFirstPageTop. 327
\HTMLLanguage: Renamed from \MetaLanguage. 356
\HTMLPageBottom: Renamed from \SetPageBottom. 327
\HTMLPageTop: Renamed from \SetPageTop. 327
General: 2017/04/29 1
lwarp-newproject removed, and combined with **lwarp**. 227
lwarpmk: Add: xdyfile configuration option. 279
lwarpmk: Fix: **xindy** and **texindy** adjusted for **pdflatex**, **xelatex** and **lualatex**. 279
lwarpmk: Fix: **xindy** now used for print index generation with **latexmk**. 279
lwarpmk: language now used for both index and glossary generation. 279
File: lwarp_html.xdy renamed to lwarp.xdy. 275
Fix: *.css files only written in print mode. 235
Fix: lwarp.xdy only written in print mode. 275
Fix: lwarp_mathjax.txt: Only written in print mode. 276
Option lwarpmklang changed to IndexLanguage. 190
Option OSWindows replaces macro \warpOSwindows. 191
Option xdyFilename added. . . . 190
Option **latexmk** replaces macro \UseLatexmk. 192
Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename. 191
v0.31
General: 2017/05/15 1
keyfloat: Improved compatibility. 750
v0.32
\RequirePackage: Fix: Ignores blanks in package list. 209
General: 2016/06/09 1
glossaries: Prevent error with \glo@name not defined. 486

lwarpmk: Fix: *io.lines()* changed to *file:lines()* due to **luatex** changes. 279
v0.33
\HTMLAuthor: Fix: Provides empty default author if none given. . . 329
\LWR@loadbefore: Fix: No \PackageError if already loaded. 196
\LWR@parseatcolumn: Fix: Column alignment with leftmost @. . . . 402
\LWR@tabledatasinglecolumntag: Fix: Macros in tabular could cause extra data cell. 416
\LWR@vspace: Add: \vspace nullified. 564
\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. 301
General: 2017/07/10 1
amsmath: Removed fleqn option. 588
fancyhdr: Fix: Optional args for \lhead, etc. 681
Add: Tabular at and bang columns now have their own HTML columns. 391
cleveref: Fix: Loaded \AtEndPreamble. 538
Fix: Incorrectly-inline math environments. 513
New handling of & to localize catcode changes. 391
v0.34
\@fnsymbol: Text symbols instead of math. 370
\InlineClass: Moved optional argument in front of mandatory. 318
\LWR@htmldivclass: Moved optional argument in front of mandatory. 316
\LWR@htmlelementclass: Moved optional argument in front of mandatory. 316
\LWR@htmlelementclassline: Moved optional argument in front of mandatory. 317
\LWR@htmlspanclass: Moved optional argument in front of mandatory. 314
\LWR@nullfonts: Improved font control. 554
\LWR@restoreorigformatting: **booktabs**: Works inside lateximage. 487
Improved font control. 487
\LWR@subhtmlelementclass: Moved optional argument in front of mandatory. 316
\LWR@tabledatacolumntag: **booktabs**: Works inside lateximage. 444
\fboxBlock: Added. 549
\makebox: Fix: Handles paren arg. . 547
General: 2017/08/08 1
babel-french: Adds fixed-width HTML spaces to punctuation. . 311
balance: Added. 609
booktabs: Works inside lateximage. 448, 613
boxedminipage2e: Added. 615
boxedminipage: Prevented. 615
crop: Added. 658
enumerate: Added. 668
enumitem: Added, no longer required. 668
everyshi: Added. 674
fancybox: Added. 676
fancyvrb: Added, no longer required. 684
figcaps: Added. 690
filecontents: Required. Patched for **morewrites**. 206
floatpag: Added. 696
flushend: Added. 702
fullpage: Added. 712
hyperxmp: Added. 739
idxlayout: Added. 741
marginfit: Added. 770
mdframed: Improved mdtheorem patch. 779
moreverb: Added. 787
paralist: Added. 813
pdflscape: Added. 816
pdfsync: Added. 819
prelim2e: Added. 822
rotfloat: Added. 831
savetrees: Added. 832
shadow: Added. 844
syntonly: Added. 866

titleps: No longer required. 880
titleref: Prevented. 884
xmpincl: Added. 938
Added. 930
Docs: Horizontal space limitations. 1
Docs: Misplaced alignment character. 176
File: `lwarp_mathjax.txt`: Version change. 276
File: `README.txt`: updated. 1
Fix: Added the `eqnarray` environments. 513
Improved font control. 552
Lists refactored to remove **enumitem** requirement. 381
Verbatim refactored to remove **fancyvrb** requirement. 376
`lateximage`: Fix: `lateximage` with `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`. . . . 522
`BlockClass`: Moved optional argument in front of mandatory. 317
`fminipage`: Added. 550
`LWR@nestspan`: Fix: Minipages, BlocksClass, and lists inside a span. 313
`LWR@tabular`: **booktabs**: Works inside `lateximage`. 449
v0.35
General: 2017/08/08 1
Fix: `\textbf` and related. 552
v0.36
`\LWR@HTMLsanitize`: Fix for **babel-french**. 518
`\LWR@HTMLsanitizeexpand`: Fix for **babel-french**. 519
`\LWR@closeparagraph`: Extra HTML source space after paragraphs. 322
`\LWR@currenttextcolor`: Fix for `\rule` when **xcolor** not loaded. 558
`\LWR@footnotetext`: Extra HTML source space after paragraphs. 331
Force HTML superscripts. 331
`\LWR@nullfonts`: Fix: Filenames while using MathJax. 554
`\LWR@restoreorigformatting`: **siunitx**: Improved super/subscripts in a `lateximage`. 487
`\LWR@section`: Improved spacing. . 348
`\LWR@stoppars`: Extra HTML source space after paragraphs. 325
`\fbox`: Fix: Uses `\fboxrule` and `\fboxsep`. 549
`\framebox`: Fix: Handles width and horiz position. 548
`\makebox`: Fix: Handles width and horiz position. 547
General: 2017/08/17 1
babel-french: Adjustements for French variants, load order, footnotes, ellipses. 311
footnote: Extra HTML source space after paragraphs. 706
siunitx: Fix for **babel-french**. . . . 530
siunitx: Improved symbol support. 848
transparent: Added. 906
upref: Added. 912
xcolor: Added `\fcolorboxBlock`, `\colorboxBlock`. 922
xcolor: Fix: Background `none` in print mode. 922
xcolor: Refactored `\LWR@colorstyle`. 926
xcolor: Uses `\fboxrule` and `\fboxsep`. 922
xcolor: `\fcolorbox` etc. now work inside `lateximage`. 922
Docs: Reorganized: Special cases and limitations. 117
Source: Improved formatting. . . . 1
`lateximage`: Footnotes appear in regular text instead of the `lateximage minipage`. 522
`LWR@tabular`: Fix for **babel-french**. 449
v0.37
`\include`: Maintains independent `aux` files for HTML. 212
General: 2017/08/19 1
LaTeX accents: Added. 225
babel-french: Adjustment for load order. 311
color: Prevented. 656
comment: Maintains independent cutfiles for print, HTML. 194

siunitx: Improved symbol support. 848
textcomp: Improved support. . . 869
lwarpmk: Removes additional HTML aux files. 279
File handles reorganized. 211
v0.38
\@seccntformat: Added for **appendix**. 347
\ForceHTMLPage: Added. 345
\ForceHTMLTOC: Added. 345
\LWR@section: \part* starts a new HTML page, for **appendix**. . . . 348
Modified spacing, uses \numberline. 348
\numberline: Added trailing \quad. 480
\part: Fix with **article** class. 353
General: 2017/08/27 1
appendix: Added. 599
arabicfront: Added. 600
caption2: Prevented. 623
chappg: Added. 626
color: Forces **xcolor** as well. . . . 656
fix2col: Added. 691
fncychap: Added. 703
grffile: Added. 729
metalogo: Added. 782
nonumonpart: Added. 798
nopageno: Added. 798
pagenote: Option page disabled. 813
realscripts: Added. 826
relsize: Added. 828
romanbarpagenumber: Added. . 831
romanbar: Added. 830
scalefnt: Added. 833
siunitx: Removed from **lwarp** core. 848
textcomp: Removed from **lwarp** core. 869
tocbibind: Added. 894
xltxtra: Added. 937
lwarpmk: Added *print1* and *html1* actions. 279
Added \markboth, \sloppy, etc. 309
Docs: Enhanced *Supported Features* table. 69
Docs: Index, **tocbibind**. 131
Docs: Starred sections. 127
v0.39
\@maketitle: **titling** version. . . . 891
Native LaTeX version. 371
Removed minipages. 371, 891
Supports **authblk** with s of class oneauthor instead of tabular. 371, 891
\AddSubtitlePublished: Added. . 372
\LWR@domulticolumn: Add: Optional vpos and # rows. 434
\LWR@restoreorigformatting: Appended with \appto instead of calling various macros. 487
\LWR@tabledatacolumntag: Don't start a data cell if see \TabularMacro. 444
\ResumeTabular: Added. 443
\TabularMacro: Added. 443
\multicolumnrow: Added. . . 441, 792
\printauthor: Removed minipages. 367
Supports **authblk** with s of class oneauthor instead of tabular. 367
\thanksmarkseries: Removed minipage footnotes. 892
General: 2017/09/05 1
a4wide: Added. 573
a4: Added. 573
a5comb: Added. 574
addlines: Added. 581
anysize: Added. 599
authblk: Added. 608
bigdelim: Added. 611
bigstrut: Added. 611
ebook: Added. 665
fullwidth: Added. 712
midpage: Added. 786
multirow: Add: New optional vpos argument. 791
multirow: Add: Supports left/right border for **bigdelim**. 791
multirow: Fix: Long text argument. 791
supertabular: Added. 865
textarea: Added. 868
titling: Improved compatibility. . 888
titling: Removed extraneous center environments. 889
typearea: Added. 909
xtabular: Added. 939
zwpagelayout: Added. 942

Docs: Reorganized tabular discussion. 153
Titlepage \published and \subtitle removed. \AddSubtitlePublished restores. 372
titlepage: Clear pending footnotes. 366
Removed minipages. 366
titlingpage: Clear pending footnotes. 889
v0.40
\@chapcntformat: Added for **tocbibind**, **anonchap**. 347
\LWR@HTMLhline: Added. 448
\LWR@includegraphicsb: Add: Full \graphicspath support. . . . 721
\LWR@nullfonts: Fix: Long arguments for expandable command. 554
\LWR@restoreorigformatting: Improved LaTeX logos inside a lateximage. 487
Improved symbols inside a lateximage. 487
Nullified \InlineClass, etc. inside a lateximage. 487
\LWR@tabledatacolumntag: Fix for **bigdelim**: \ldelim, \rdelim. . 444
\chapter: Added support for **quotchap**. 353
\multicolumnrow: Fix: Adapts to older **multirow** and **xparse**. . . . 441
\simplechapterdelim: Added for **tocbibind**, **anonchap**. 347
\underline: Added. 558
General: 2017/09/25 1
adjmulticol: Added. 580
anonchap: Added. 599
bigdelim: Improved documentation. 611
cuted: Added. 658
dblfnote: Added. 660
fnpos: Added. 704
graphics: Moved out of the **lwarp** core. 716
graphics: Restores \includegraphics and \DeclareGraphicsExtensions in a lateximage. 716
graphicx: Moved out of the **lwarp** core. 728
grffile: Directly supported. 729
midfloat: Added. 786
multirow: Improved **bigdelim** borders. 791
pfnote: Added. 821
quotchap: Added. 824
sectsty: Added. 842
stabular: Added. 858
tabls: Added. 867
textcomp: Additional symbols, improved XeLaTeX and LuaLaTeX support. 869
tocbibind: Improved for \simplechapter. 894
xltxtra: Fix for \showhyphens with XeLaTeX. 937
No longer preloads **xfrac**. 207
v0.41
\LWR@addcmidruletrim: Add: \cmidrule trims. 422
\LWR@clearmidrules: Add: \cmidrule trims. 419
\LWR@closetabledatacell: Add: Mute > for \bottomrule. . . . 396
Fix: At/bang column with \multirow. 396
Fix: Cancel < for \multicolumn. 396
\LWR@domulticolumn: Add: \cmidrule trims. 434
Added vertical rules. 435
\LWR@nullifyNoAutoSpacing: Fix: \NoAutoSpacing in a tabular with **babel-french**. 449
\LWR@parsebarcolumn: Added vertical rules. 404
\LWR@printatbang: Add: \cmidrule trims. 415
Add: Mute at and bang columns for \bottomrule. 415
\LWR@printbartag: Added vertical rules. 415
\LWR@subaddcmidruletrim: Added. 422
\LWR@subcmidrule: Add: \cmidrule trims. 420
\LWR@tabledatasinglecolumntag: Add: \cmidrule trims. 416
Add: Mute < for \bottomrule. . . 416

\LWR@tabularfinishrow: Unfinished tabular rows automatically filled. 398
\mcolrowcell: Added for \multicolumrow cells. 447
General: 2017/10/07 1
multirow: Add: \cmidrule trims. 791
Added vertical rules. 791
Fix: < spec. 792
Improved rules. 613
LWR@tabular: Fix: \NoAutoSpacing in a tabular with **babel-french**. 450
v0.42
\@ensuredmath: Improved \ensuremath. 503
\@textsubscript: Added. 558
\@textsuperscript: Added. 557
\LWR@HTMLhline: If FormatWP force explicit border. 448
\LWR@HTMLtextstyle: Added. 552
\LWR@addformatwpalignment: If FormatWP add explicit style for cell alignment. 424
\LWR@addrulewidth: If FormatWP force explicit border. 423
\LWR@amsmathbody: Fix: Numbering and naming AMS math environments. 521
\LWR@amsmathbodynumbered: Fix: Numbering and naming AMS math environments. 522
\LWR@doequation: If FormatWP print LaTeX expression. 508
\LWR@domulticolumn: If FormatWP add cell alignment. 435
\LWR@doubledollar: If FormatWP print LaTeX expression. 501
Improved \ensuremath. 501
Improved line spacing with mathjax. 501
\LWR@floatbegin: If FormatWP add a text frame. 467
\LWR@floatend: If FormatWP add a text frame. 468
\LWR@htmlmathlabel: If FormatWP print LaTeX expression. 512
\LWR@includegraphicsb: Fix: Filename expansion. 721
If FormatWP, use explicit size. . . . 722
\LWR@remembertag: Fix: Numbering and naming AMS math environments. 521
\LWR@restoreorigformatting: Improved \ensuremath. 489
\LWR@subaddcmidruletrim: 0pt if no rule given. 422
\LWR@subsingledollar: If FormatWP print LaTeX expression. 495
\LWR@tabledatasinglecolumntag: If FormatWP add cell alignment. 417
\LaTeX: If FormatWP use explicit style. 569
\TeX: If FormatWP use explicit style. 568
\hspace: If FormatWP add \quads. . 563
\listoffigures: Added boolean WPMarkLOFT. 477
\listoftables: Added boolean WPMarkLOFT. 477
\marginpar: If FormatWP emulate a wrapfig. 335
\marginparBlock: If FormatWP emulate a wrapfig. 335
\rule: If FormatWP add \quads. . . . 567
\tableofcontents: Added boolean WPMarkTOC. 477
\underline: If FormatWP, use explicit styles for \underline, etc. 558
General: 2017/10/30 1
\textbf and related: If FormatWP, use explicit styles for \textsc, etc. 552
algorithmicx: If FormatWP add \quads. 586
epigraph: If FormatWP add HTML styles. 669
fancybox: If FormatWP add HTML styles. 676
floatflt: Added width. 695
includegraphics: Fix: Class key. . 720
keyfloat: If FormatWP add explicit HTML style. 751
moreverb: Simplified formatting of listings. 787
morewrites: Added. 788
multirow: If FormatWP add cell alignment. 791
overpic: Added. 812

realscripts: Fix for subscripts in a `lateximage`. 826
sidenotes: If `FormatWP` add explicit HTML style. 846
siunitx: Improved `\ensuremath`. 849
soul: If `FormatWP`, add explicit styles. 853
textcomp: Improved `\interrobangdown`. 869
wrapfig: If `FormatWP` add explicit HTML style. 921
Added boolean `WPMarkLOFT`. . . . 221
Added boolean `WPMarkMath`. . . . 221
Added boolean `WPMarkMinipages`. 220
Added boolean `WPMarkTOC`. 220
Added boolean `WPTitleHeading`. 221
Docs: Added support page. 2
Docs: Improper `\prevdepth`. . . . 176
Docs: Reorganized math limitations 144
File: `lwarp_mathjax.txt`: Updated **siunitx** script. 276
Fix: Numbering and naming AMS math environments. 520
If `FormatWP` force explicit border. 613
If `FormatWP`, shift section headings. 221
`tabbing`: Added. 379
`lateximage`: Fix: Numbering and naming AMS math environments. 523
`center`: If `FormatWP` use explicit `text-align`. 528
`minipage`: Added boolean `WPMarkMinipages`. 545, 546
If `FormatWP` add a text frame. . . . 544
`eqnarray`: Fix: Numbering and naming AMS math environments. 515
If `FormatWP` print LaTeX expression. 514
`LWR@BlockClassWP`: Added to factor code. 318
`LWR@figcaption`: If `FormatWP` forces italic captions. 471

v0.43

`\LWR@domulticolumn`: Fix for vertical rules. 435
Fix: Multicolumn trim. . . . 434, 435
`\LWR@maybeprintpendingfootnotes`: Added `FootnoteDepth`. 334
`\LWR@nullfonts`: Fix: Nullify dollar inside filesnames. 554
`\LWR@parsetablecols`: Ignore spaces in col spec. 409
`\LWR@printmccoltype`: Added vertical rules. 430
`\LWR@section`: Fix: Expansion in comparison. 348
Fix: Math in section name. 350, 352
Fix: Nullify fonts inside HTML comment. 349
`\TabularMacro`: `\newcommand` instead of `\relax` to fix **supertabular** and **xtab**. 443
`\href`: Made robust. 463
`\nameref`: Made robust. 461
`\nolinkurl`: Made robust. 463
`\url`: Made robust. 464
General: 2017/11/08 1
breakurl: Added. 616
hyperref: Made robust. 734, 736, 738
hyperref: `\Gauge` added. 738
luatodonotes: Added. 767
todonotes: Added. 904
`LWR@currentautosec`: Added. . . 348
Added `FootnoteDepth`. 330
Docs: HTML settings table. . . . 106
Docs: Reorganized HTML customization. 106

v0.44

`\@currentlabelname`: Adjustment for **koma-script**. 454
`\HTMLTitle`: Added. 328
`\LWR@addformatwpalignment`: Fix for multicolumn alignment if `FormatWP`. 424
`\LWR@backgroundcolor`: Added. . . 926
`\LWR@filestart`: Add `\HTMLTitle`. 357
Fix `\HTMLAuthor`. 356
`\LWR@listitem`: Added `list` and `trivlist`. 384
`\LWR@patchlists`: Added `list` and `trivlist`. 387
`\LWR@strresult`: Fix: `\providecommand`. 394
`\LWR@textcurrentcolor`: **xcolor**: Added `\LWR@textcurrentcolor`. . . 925

\addcontentsline: Automatic \LWR@newfloatanchor. 473
\chapter: Add preamble for **koma-script**. 353
\marginparBlock: Added. 335
\nopagecolor: **xcolor**: Fix for \nopagecolor. 927
\part: Add preamble for **koma-script**. 353
\title: Added \thetitle. 328
General: 2017/11/22 1
algorithmicx: Improved comment symbol. 586
atbegshi: Added. 603
cancel: Added. 619
changepage: Additional options. 626
easy-todo: Added. 664
fancyref: Added. 682
fixmetodonotes: Added. 692
fixme: Added. 691
fontenc: Allowed after **lwarp**. . . . 704
hang: Added. 730
ifoddpage: Added. 741
ltxtable: Added. 766
luatodonotes: Improved. 767
lwarp-patch-komascript: Added. 944
overpic: Fix: Groups for lateximages. 812
pdfsync: Fixes. 819
preview: Added. 822
scrextend: Added. 834
scrhack: Added. 837
scrlayer-notecolumn: Added. . . . 839
scrlayer-scrpage: Added. 839
scrlayer: Added. 837
section: Added. 840
soulpos: Added. 855
soulutf8: Added. 856
supertabular: Fix for caption. . . 866
tikz: Fix: Groups for lateximages. 879
tocbasic: Added. 893
tocloft: Added \newlistentry. . 901
tocloft: Improved \newlistof. . 901
tocstyle: Added. 902
todonotes: Improved. 904
todo: Added. 903
typearea: Added expert commands. 909
watermark: Added. 920
xcolor: Added \LWR@currenttextcolorstyle. 925
xcolor: Added \LWR@findcurrenttextcolor. 925
xtab: Fix for caption. 940
Adjustment for **koma-script**. . . . 203
AMS environments: Fix: Groups for lateximages. 588
If pdfLaTeX, require T1 and UTF8 encoding. 185
picture: **overpic**: Fix: Groups for lateximages. 541
list: Added list and trivlist. . 384
LWR@nestspan: Added list and trivlist. 313
v0.45
\@currentHref: Added. 462
\@donoparitem: Modified for HTML. 381
\@item: Modified for HTML. 382
\@mklab: Modified for HTML. 381
\CSSFilename: Improved filenames with underscores. 328
\LWR@LwarpStart: Fix: Lateximages on incorrect pages with Mathjax. 360
\LWR@filenamenoblanks: Fix: Section names with \ 338
Fix: Section names with underscores. 338
\LWR@includegraphicsb: Improved URLs with underscores. 721
\LWR@newautoidanchor: Fix: No anchor if frozen autoid. 469
\LWR@notmemoirloadafter: Added. 196
\LWR@printpendingmpfootnotes: Added. 334
\LWR@startref: Fix: Labels with underscores. 460
\LWR@subhyperref: Improved URLs with underscores. 462
\LWR@subhyperrefclass: Improved URLs with underscores. 463
\LWR@sublabel: Fix: Labels with underscores. 457
\LWR@tabledatacolumntag: Fix: Empty line between rows. 447
\chapter: Add optional heading title for **memoir**. 353
\newpage: Added. 561
\nolinkurl: Fix: Underscore in URL. 463
\normalmarginpar: Added. 336

\reversemarginpar: Added. 336
\section: Add optional heading title for **memoir**. 354
\tableofcontents: Fix: Empty sidetoc. 476
Fix: Patch \AtBeginDocument. . . 476
\url: Improved URLs with underscores. 464
General: 2018/01/14 1
array: Added. 600
babel-french: Robust commands. 311
backref: Added. 609
breakurl: Fix: Underscore in URL. 616
changebar: Added. 625
cite: Added. 655
continue: Added. 658
endfloat: Added. 666
epigraph: Support for **memoir**. . 669
fancyvrb: Improvements. . 684, 685
flafter: Added. 693
fltrace: Added. 702
footnpag: Added. 708
fwlw: Added. 712
hanging: Added. 732
hyperref: Fix: Underscore in URL. 734, 735
lwarp-patch-memoir: Added. . . . 947
memhfixc: Added. 782
memoir: Added. 571
natbib: Added. 794
pagesel: Added. 813
prettyref: Added. 822
subfigure: Added. 864
subfig: Fix for subcaption end tag. 862
subfig: Fix: Math in subcaptions. 860
textfit: Added. 872
titleref: Added. 884
turnthepage: Added. 908
Allows **memoir**'s preloaded packages. 197
Docs: **xparse** warnings. 162
Docs: Fix for double hyphens. . . 83
Docs: Improved install instructions. 83
Docs: Improved MiKTEX install instructions. 79
Docs: Moved table so doesn't interfere with install docs. 78
File: lwarp_mathjax.txt: Allow MathJax inside tabbing. . . . 276
File: lwarp_mathjax.txt: Allow MathJax inside verse. 276
Fix: Empty sidetoc. 476
Improved: Robust \, \ , and \textellipsis commands. . . 559
Separate LWR@thisautoidWP for word processor s. 469
thebibliography: Patched to emphasize titles. 486
minipage: Fix: Improper \prevdepth. 543, 546
v0.46
\LWR@closeparagraph: Fix: Tabular empty lines. 323
\LWR@closeprevious: Fix: Stack unnesting. 309
\LWR@forcenewpage: Fix: Improper \prevdepth. 309
\LWR@lookforpackagename: Fix: Spaces in \usepackage. 209
\LWR@providelength: Added. 186
\LWRPrintStack: Name changed from \PrintStack. 308
\makebox: Fix: Lateximage in a \makebox. 548
\popclose: Fix: Stack unnesting. . . 299
\pushclose: Fix: Stack unnesting. . 299
General: 2018/01/23 1
LWR@tabularpardepth added. . . 394
amsthm: Adapted to trivlist changes. 596
mdframed: Fixes for SVG math or lateximage in title. 775
mdframed: Fixes for footnotes. . 776
ntheorem: Adapted to trivlist changes. 799
theorem: Adapt to trivlist changes. 875, 877
list: Fix: Stack unnesting. 384
LWR@tabular: Fix: Tabular empty lines. 452, 453
v0.47
\LWR@HTML@caption@begin: Fix: Argument passed to \LWR@origcaption@begin. . . 472
\LWR@LwarpStart: Fix for SVG math in \nameref. 361
\LWR@WPcell: Fix: Line wrap at HTML hyphen. 424

\LWR@createautosec: Fix: Line wrap at HTML hyphen. 346
\LWR@domulticolumn: Fix: Line wrap at HTML hyphen. 435
\LWR@floatbegin: Fix: Line wrap at HTML hyphen. 467
\LWR@htmlclosecomment: Add \mbox to prevent line breaks. . 314
\LWR@newautoidanchor: Fix: Line wrap at HTML hyphen. 469
\LWR@printopenlist: Fix: Line wrap at HTML hyphen. 381
\LWR@startref: Fix: Line wrap at HTML hyphen. 459
\LWR@sublabel: Fix: Line wrap at HTML hyphen. 458
\LWR@subsingledollar: Added SVG math image baseline adjust and em sizing. 495
Fix: Line wrap at HTML hyphen. . 499
\captionlistentry: Fix: Line wrap at HTML hyphen. 473
\hypertoc: Fix: Line wrap at HTML hyphen. 480
\hypertocfloat: Fix: Line wrap at HTML hyphen. 481
\makebox: Fix: Line wrap at HTML hyphen. 547
General: 2018/01/30 1
adjmulticol: Fix: Line wrap at HTML hyphen. 580
blowup: Added. 612
caption: Added. 620
caption: Also loads **lwarp-caption**. 211
changepage: Fix for pagecheck macros. 626
endheads: Added. 666
epigraph: Fix: Line wrap at HTML hyphen. 669
hanging: Fix: Line wrap at HTML hyphen. 732
hang: Fix: Line wrap at HTML hyphen. 730
keyfloat: Fix for SVG math in captions. 750
midpage: Fix: Line wrap at HTML hyphen. 786
multirow: Fix: Line wrap at HTML hyphen. 791
multitoc: Added. 793
ntheorem: Fix: Line wrap at HTML hyphen. 804
realscripts: Fix: Line wrap at HTML hyphen. 826
scrextend: Fix: Line wrap at HTML hyphen. 834
sectionbreak: Added. 841
sidenotes: Fix for SVG math in captions. 846
subfig: Fix for SVG math in captions. 860
subfig: Fix: Support \nameref. . . 859
xurl: Added. 941
lwarpmk: **pdfcrop**: Removed hires option for improved crop accuracy. 279
lateximage: Added CSS style option. 522
Fix: Line wrap at HTML hyphen. . 527
center: Fix: Line wrap at HTML hyphen. 528
minipage: Fix: Line wrap at HTML hyphen. 544
flushleft: Fix: Line wrap at HTML hyphen. 528
flushright: Fix: Line wrap at HTML hyphen. 528
enumerate: Fix: Line wrap at HTML hyphen. 385
itemize: Fix: Line wrap at HTML hyphen. 385
LWR@BlockClassWP: Fix: Line wrap at HTML hyphen. 318
v0.48
\@@@setcpageref: Fix for new v0.21 of **cleveref**. 539
\@@@setcref: Fix for new v0.21 of **cleveref**. 538
\@@@setcrefrange: Fix for new v0.21 of **cleveref**. 539
\@biblabel: Improved bibliography label. 486
\@item: Honors \makelabel. . . . 382
\@maketitle: Fix: Errors with **IEEEtran** class. 371
\LWR@LwarpStart: Adjusted space around captions. 359
\LWR@ProvidesPackageDrop: Fix: Options with braces. 211

\LWR@addtabularhrulecolor: **colortbl**: Added. 425
\LWR@addtabularrulecolors: **colortbl**: Added. 426
\LWR@closetabledatacell: **colortbl**: Added. 396, 397
\LWR@includegraphicsb: Fix: Virtual page size limited to a group. 721, 724
\LWR@lookforpackagename: Fix: Parsing similar package names. 208
\LWR@newautopagelabel: Fix: TOC, LOF, LOT links. 340
\LWR@newhtmlfile: Fix: TOC, LOF, LOT links. 343
\LWR@nullfonts: Fix: \newline in title. 554
\LWR@parsedrequirepackagenames: Fix: Parsing similar package names. 208
\LWR@parsetablecols: Fix: Ignore optional tabular column arguments. 410
\LWR@restoreorigformatting: Fix: Spacing in SVG math, lateximage, Ti*k*z. 488
\LWR@section: Fix: TOC, LOF, LOT links. 352
\LWR@tabledatasinglecolumntag: **colortbl**: Added. 418
\LWR@textcurrentfont: Added. Improves font control. 556
\bibliography: Fix: \BaseJobname for bibliography. 486
\centerline: Added. 529
\l@part: Adapts to classes without \part. 482
\leftline: Added. 529
\mbox: Nullified for HTML. 547
\rightline: Added. 529
\thempfootnote: Removed \itshape. 334
General: 2018/02/14 1
acronym: Added. 578
acro: Added. 577
chapterbib: Added. 626
colortbl: Added. 412, 425, 656
fancyheadings: Prevented. 681
fancyref: Now directly supported. 682
hypcap: Added. 733
hypernat: Added. 733
hyperref: \texorpdfstring now uses the TEX string. 738
luatodonotes: Improved \todototoc. 767
siunitx: Changes fraction to symbol. 850, 851
siunitx: Improved SVG math. 849, 850
siunitx: Improved color output. . 849
stfloats: Added. 859
todonotes: Improved \todototoc. 904
vmargin: Added. 915
xfrac: Fix: Added groups around super/subscripts to localize LWR@nestspan changes. 935
Docs: Converting an existing document. 99
Improved font control. 556
lateximage: Print mode boxed to natural width. 528
abstract: Allow optional name. . . 373
LWR@tabular: **colortbl**: Added. . . . 451
v0.49
\LWR@addtabularcellcolor: **xcolor**: Added tabular row colors. 428
\LWR@domulticolumn: **xcolor**: Added tabular row colors. 435
\LWR@printlength: Fix: Group **printlen** changes. 207
\affiliation: Fix: Adapts to classes which already provide. 365
\href: Fix: Adapt to classes. 463
\noalign: Fix: \noalign inside tabular. 448
\url: Fix: Adapt to classes. 464
General: 2018/02/19 1
amsmath: Fix: Patches for \eqref. 588
eso-pic: Fix for \AddToShipoutPicture. . . . 673
figsize: Added. 690
fnlineno: Added. 704
hypdestopt: Added. 733
hyphenat: Added. 740
lineno: Added. 754
luacolor: Added. 766
pagegrid: Added. 813
pdfrender: Added. 819
resizegather: Added. 830

vertbars: Added. 915
vwcol: Added. 917
xcolor: Added tabular row colors. 412, 931
Fix: Adapt to classes. 560
v0.50
`\@ensuredmath`: Fix: Use `lateximage` even if MathJax. . 503
Improved SVG math `alt` tags. . . . 503
`\LWR@atbeginverbatim`: Improved column alignment. 377
`\LWR@doequation`: Improved SVG math display. 508
`\LWR@doubledollar`: Improved SVG math `alt` tags. 502
Improved SVG math display. 502
`\LWR@footnotetext`: Robustify macros. 332
`\LWR@htmlrefsectionfilename`: Fix: SVG math in a section name. 307
`\LWR@newhtmlfile`: Fix: SVG math in a section name. 342
`\LWR@nullfonts`: Fix: `\underline` in sectioning file name. 555
Robustify macros. 555
`\LWR@overline`: Added. 558
`\LWR@subsingledollar`: Fix: Use `lateximage` even if MathJax. . 495
Improved SVG math `alt` tags. . . . 495
MD5 hash avoids duplicate SVG math. 500
`\LWR@vspace`: Robustify macros. . . 564
`\newline`: Robustify macros. 561
`\textsubscript`: Robustify macros. 558
`\textsuperscript`: Robustify macros. 557
General: 2018/03/03 1
`lwarp.css`: Improved SVG display math centering. 235
`lwarp_one_limage.txt`: Added. 276
amsmath: Fix: Upright tags for SVGmath. 588
axodraw2: Added. 609
bytefield: Added. 619
dblfloatfix: Added. 660
diagbox: Added. 661
epstopdf: Added. 670
listings: Force flexible columns. . 757
morefloats: Added. 786
nonfloat: Added. 798
ntheorem: Fix: Not `standard` nor `amsthm` selected. 806
pbox: Added. 815
phfqit: Added. 821
schemata: Added. 833
siunitx: Fix: Loads **xcolor**. 848
siunitx: Improved SVG math `alt` tags. 850
siunitx: Improved units. 530, 848, 851
xy: Added. 941
lwarpmk: Error if `lateximages.txt` does not exist. 279
lwarpmk: Error if `lwarpmk.conf` points to **lwarp**. 279
lwarpmk: Improved error messages. 279
lwarpmk: MD5 hash avoids duplicate SVG math. 279
lwarpmk: Multiprocess support making lateximages. 279
AMS environments: Improved SVG math display. 588
Fix: Load **fontspec** if necessary. . 204
Robustify macros. 556
`lateximage`: Fix: SVG math in a section name. 525
MD5 hash avoids duplicate SVG math. 524, 527
`eqnarray`: Improved SVG math display. 515
v0.51
`\@ensuredmath`: Hashes `\ensuremath`. 503
`\@item`: Restored list label space. . . 383
`\LWR@HTMLsanitize`: Fix: Escapes double quotes. 518
`\LWR@HTMLsanitizeexpand`: Fix: Escapes double quotes. 519
`\LWR@LwarpStart`: MathJax: Nullifies `\ensuremath`. 361
`\LWR@addbaselinemarker`: Improved SVG math baseline. . 494
`\LWR@atbeginverbatim`: Adds vertical offset. 377
`\LWR@customizeMathJax`: MathJax: Nullifies `\ensuremath`. 340
`\LWR@doequation`: Fix: `\addcontentsline` inside SVG math. Provides an autoid anchor. 508

\LWR@doubledollar: Fix: \addcontentsline inside SVG math. Provides an autoid anchor. 502
\LWR@findcurrenttextcolor: Added \LWR@findcurrenttextcolor when no **xcolor**. 559
\LWR@newautoidanchor: Fix: No autoid is inside a lateximage. 469
\LWR@newhtmlfile: MathJax: Nullifies \ensuremath. 344
\LWR@subsingledollar: Fix: \ensuredmath inside SVG image. 496
Fix: lateximage inside $\mathcal{AMS}$ \text. 497
Fix: Honors text font around SVG math. 497
Fix: SVG math with enclosed lateximage. 496
Improved SVG math baseline. . . . 498
SVG math baseline improved with invisible rule at corner. 500
Typeset SVG math only once during measurement. 497
\LWR@textcurrentcolor: **xcolor**: \LWR@textcurrentcolor if **xcolor** not loaded. 559
\addcontentsline: Add missing support for float mechanism if necessary. 474
No anchor ID if inside SVG image. 473
\displaymathnormal: Processing for complicated display math. . . . 511
\displaymathother: Processing for complicated display math. . . . 511
\textcolor: Fix: SVG math color. . 927
General: 2018/03/24 1
lwarp_one_limage.txt: **pdftocairo** -noshrink added. . 276
afterpackage: No longer required. 205
chemfig: Added. 627
chemformula: Added. 628
chemgreek: Added. 633
chemmacros: Added. 634
chemnum: Added. 654
epstopdf-base: Added. 670
fancybox: Fix: Optional tag for \item in a span. 679
grid: Added. 729
listings: Forces cleared options. . 758
ltxgrid: Added. 765
mhchem: Added. 783
tikz: Fix for \tikz macro. 879
tikz: Fix for tikz with optional argument. 879
titling: Fix for \thanks mark. . . . 890
lwarpmk: **pdfcrop**: Restored hires option. 279
lwarpmk: **pdftocairo** -noshrink added. 279
AMS environments: Fix: \addcontentsline inside SVG math. Provides an autoid anchor. 588
Docs: **tikz** limitations. 151
Docs: Multiple authors and affiliations. 127
Docs: Things to avoid. 118
Docs: Updated Converting an existing document. 99
Fix: Remember original \# in case is redefined. 223
HTML entity used for text dollar. 494
lateximage: Added additional hashing option. 522
Fix: lateximage inside $\mathcal{AMS}$ \text. 523
Processing for complicated display math. 525
alignat: Fix: Added. 593
eqnarray: Fix: \addcontentsline inside SVG math. Provides an autoid anchor. 515
LWR@displaymathother: Processing for complicated display math. . 504
LWR@equationother: Processing for complicated display math. . . 505
v0.52
\@ensuredmath: Improved hashing expansion. 503
\@mpfootnotetext: Fix: Paragraph handling. 333
\CustomizeMathJax: Added. 340
\LWR@addbaselinemarker: Warnings if lwarp_baseline_marker.png is not present or if graphicx/s not loaded. 494
\LWR@customizedMathJax: Added. 340

\LWR@doequation: Fix: equation* now based on equation instead of displaymath. 508
Fix: equation* with split. 508
\LWR@filenamenoblanks: Fix: \FileDepth with non-utf8 encoding. 339
\LWR@footnotetext: Fix: Paragraph handling. 332
\LWR@newhtmlfile: Fix: \FileDepth with non-utf8 encoding. 343
\LWR@nullfonts: Fix: \texorpdfstring in section names. 555
\LWR@section: Fix: Footnote numbering: Limited HTML comment if starred. 349
Fix: Footnote numbering: Use short TOC entry for HTMLDebug comments. 349
\LWR@subsingledollar: Added user-adjustable SVG math font scaling. 498
\LateximageFontScale: Added user-adjustable SVG math font scaling. 518
\href: Fix: #, %, &, ~, _ in URL. 463
\nolinkurl: Fix: #, %, &, ~, _ in URL. 463
\url: Fix: #, %, &, ~, _ in URL. 464
General: 2018/04/01 1
breakurl: Fix: #, %, &, ~, _ in URL. 616
endfloat: Updated for v2.6. 666
fancyvrb: Initial support for \VerbatimFootnotes. . . . 676, 684
hyperref: Fix: #, %, &, ~, _ in URL. 734–736
nicefrac: Added. 797
url: Added. 913
lwarpmk: Fix: Memory overflow when spawning tasks. 279
lwarpmk: Fix: Skip image generation if from page 0. 279
Changed FootnoteDepth default to \subsbusection. 330
Docs: Improved install instructions. 80
Fix: MathJax script line wraps. Reduced right margin. 205
If pdfLaTeX, allow other input encoding. 185
pkggraphics: Added defaults. 718, 719
pkggraphicx: Updated for v1.1a. . 719
pkggraphicx: Updated for v1.1b. . 719
Restore \kill in a lateximage. . 763
tabbing: Fix to allow inside lateximage. 379
lateximage: Fix for hash expansion. 524
v0.53
General: 2018/04/01 1
lwarpmk: Added ***lwarpmk cleanlimages***. 279
lwarpmk: Added warning for corrupted images. 279
Docs: ***lwarpmk cleanlimages***. . 97
Docs: ***lwarpmk pdftohtml***. . . . 97
v0.54
\LWR@afterendverbatim: Added vspace argument. 378
\LWR@atbeginverbatim: Improved column alignment. 377
\LWR@earlyloadnever: Added. . . . 196
\LWR@endfloatalignment: Honor \centering, etc. in floats. . . . 470
\LWR@floatalignment: Honor \centering, etc. in floats. . . . 470
\LWR@floatbegin: Honor \centering, etc. in floats. . . . 468
\LWR@floatend: Honor \centering, etc. in floats. 468
\LateximageFontSizeName: Defaults to normalsize. 518
\centering: Added debug comment. 529
\raggedleft: Added debug comment. 529
\raggedright: Added debug comment. 529
General: 2018/04/22 1
*.lwarpmkconf: Option IndexLanguage changed to xindyLanguage. 234
*.lwarpmkconf: Option pdftotextEnc added. 234
*.lwarpmkconf: Option xdyFilename changed to xindyStyle. 234
*.lwarpmkconf: Option xindyCodepage added. 234

lwarp.css: Fix: Text-decoration-skip: auto. . . . 235
lwarpmk.conf: Option IndexLanguage changed to xindyLanguage. 234
lwarpmk.conf: Option pdftotextEnc added. 234
lwarpmk.conf: Option xdyFilename changed to xindyStyle. 234
lwarpmk.conf: Option xindyCodepage added. 234
aecc: Prevented. 581
ae: Prevented. 581
bibunits: Added. 610
chngpage: Added. 626
forest: Added. 708
glossaries: Fix when not using **babel** or **polyglossia**. 715
gridset: Added. 730
hyperref: Fix: \hyperref and \hyperlink with special chars in text. 736
hyperref: Fix: \ref in \hyperref and \hyperlink caused nested link. 736
inputenx: Added. 749
lwarp-patch-memoir: Update for v3.7g. 951
magaz: Added. 769
ragged2e: Fix: \centering, etc. . 825
t1enc: Prevented. 867
textcomp: Fix for \textperthousand. 869
tikz: Fixes for \pgfpicture, minipages, fit, align, font. . . . 879
wasysym: Prevented. 920
lwarpmk: Added pdftotextenc. 279
lwarpmk: Added xindycodepage. 279
lwarpmk: Changed language to xindylanguage. 279
lwarpmk: Changed xdyfile to xindystyle. 279
lwarpmk: Improved error if configuration file does not exist. 279
lwarpmk: Increased prominence for error for an unknown command. 279
lwarpmk: Verifies HTML version exists before lwarpmk limages. 279
lwarpmk: Verifies image references before lwarpmk limages. 279
Add: pdftotextEnc. 190
Add: xindyCodepage. 190
Added early check for disallowed packages. 197
Docs: BibTeX. 129
Docs: Macros in sectioning names. 118
Option IndexLanguage changed to xindyLanguage. 190
Option xdyFilename changed to xindyStyle. 190
verse: Fix: Line spacing. 375
v0.55
\LWR@LwarpStart: Fix: Overfull boxes in lateximages. 359
\LWR@floatbegin: Fix: Float optional args. 468
\LWR@nullfonts: Removed extraneous space which appeared in file links. 555
\phantomsection: Fix: \ForceHTMLTOC with \phantomsection. 568
General: 2018/04/26 1
clrdblpg: Added. 655
Fix: \centering, etc. for **koma-script**. 468
Fix: QED symbols in lateximage. 598, 811
v0.56
\LWR@addcdashline: **arydshln**: Added. 424
\LWR@addmulticolvertrulecolor: Adds support for dashed vertical rules. 433
Adds support for double vertical rules. 433
\LWR@addtabularhrulecolor: Adds support for **arydshln** dashed rules. 425
Adds support for double \hlines and \midrules. 425
\LWR@addtabularrulecolors: Adds support for dashed vertical rules. 426
Adds support for double vertical rules. 426

\LWR@closeparagraph: Added support for **parnotes**. 323
\LWR@domulticolumn: Adds support for dashed vertical rules. 435
Adds support for double vertical rules. 435
\LWR@floatbegin: Adds a <class> per **float** package style. 467
\LWR@multicolpartext: Fix: \multicolumn parameters. . . . 430
\LWR@openparagraph: Added support for **parnotes**. 322
\LWR@parsebarcolumn: Adds support for double vertical rules. 404
\LWR@parsecoloncolumn: **arydshln**: Added. 405
\LWR@parsesemicoloncolumn: **arydshln**: Added. 405
\LWR@parsetablecols: Added **array** W column. 411
\LWR@printmccoldata: Added **array** W column. 431
Fix: \multicolumn parameters. . 431
\LWR@printmccoltype: Added **array** W column. 429
Adds support for dashed vertical rules. 430
Adds support for double vertical rules. 430
\LWR@tabledatacolumntag: Fix: \morecmidrules 446
\LWR@textcurrentfont: Added span.textbf, etc. 556
General: 2018/05/12 1
*.lwarpmkconf: Records -shell-escape. 234
lwarp.css: Added div.textbf, etc. 235
lwarp.css: Added span.textbf, etc. 235
lwarpmk.conf: Records -shell-escape. 234
arydshln: Added. 391, 601
lua-check-hyphen: Added. 766
paralist: Fixes for compactenum, compactitem, compactdesc. . . 813
parnotes: Added. 814
quoting: Added. 825
tocenter: Added. 895
underscore: Added. 912
lwarpmk: Added lwarpmk pdftosvg. 279
lwarpmk: Supports -shell-escape. 279
Added \thinspace. 559
Docs: ***lwarpmk pdftosvg***. 97
LWR@blocktextcurrentfont: Added div.textbf, etc. 556
v0.57
\BlockClassSingle: Improved print/HTML output selection. . 318
\InlineClass: Improved print/HTML output selection. . 318
\LWR@customizeMathJax: MathJax: Supports \footnote, \footnotemark. 340
\LWR@ref@ignorestar: **subcaption**: Fix: \subref. 461
\LWR@subhyperref: Fix: Text catcodes. 462
\LWR@subhyperreftext: Fix: Text catcodes. 463
\LWR@subsingledollar: Fix: Dynamic inline math expressions. 496, 497, 500
\LWR@vspace: Improved print/HTML output selection. 564
\StartDefiningMath: Added. . . . 302
\StartDynamicMath: Added. 303
\boxframe: **xcolor**: Fix: Colored \boxframe. 931
\colorbox: **xcolor**: New system for switching print and HTML outputs. 928
\colorboxBlock: **xcolor**: New system for switching print and HTML outputs. 928
\fboxBlock: Improved print/HTML output selection. 549
\fcolorbox: **xcolor**: New system for switching print and HTML outputs. 928
\framebox: Improved print/HTML output selection. 548
\href: Fix: Text catcodes. 463
\listof: Fix: Provide \l@name if not defined. 478
\makebox: Improved print/HTML output selection. 547
\mathimagename: Added. 493

\mbox: Improved print/HTML output selection. 547
\multicolumnrow: **multirow**: Improved print/HTML output selection. 792
Improved print/HTML output selection. 441
\newfloat: **rotfloat**: Added float styles. 832
rotfloat: Fix for listof sideways floats. 832
\packagediagramname: Added. . . . 493
\parbox: Improved print/HTML output selection. 547
\raisebox: Improved print/HTML output selection. 551
\reflectbox: Improved print/HTML output selection. 728
\resizebox: Improved print/HTML output selection. 728
\rotatebox: Improved print/HTML output selection. 726
\rule: Fix: Colored rules. 566
\scalebox: Improved print/HTML output selection. 727
\textcolor: **xcolor**: New system for switching print and HTML outputs. 927
General: 2018/06/06 1
lwarp.css: Added ruled, boxed, boxruled floats. 235
lwarp.css: Increased float vertical margins. 235
algorithm2e: Added. 582
bigdelim: Improved print/HTML output selection. 611
breakurl: Fix: Text catcodes. . . . 616
colortbl: New system for switching print and HTML outputs. . 656, 657
ellipsis: Added \midwordellipsis. 665
errata: Added. 671
float: Added float styles. 694
float: Fix: Do not pre-define \l@name. 694
glossary: Prevented. 716
ltablex: Added. 764
marginnote: Fix: Long optional argument. 770, 771
multirow: Improved print/HTML output selection. 791
register: Added. 827
subcaption: Fix: \subref. 735
trimclip: Added. 906
vowel: Added. 916
xellipsis: Added. 933
xfrac: Improved print/HTML \scalebox control. 935
xltabular: Added. 937
xpiano: Added. 938
lwarpmk: Improved code factoring. 279
lwarpmk: Improved error handling. 279
Docs: Recompiling **lwarpmk** or css files. 175
Docs: Recreating the index for **lwarp** source. 171
New system for switching print and HTML outputs. 217
minipage: Improved print/HTML output selection. 543
BlockClass: Improved print/HTML output selection. 317
fminipage: Improved print/HTML output selection. 550
LWR@BlockClassWP: Improved print/HTML output selection. . 318
v0.58
\LWR@HTML@caption@begin: Improved print/HTML output selection. 472
\LWR@HTML@caption@end: Improved print/HTML output selection. . 472
\LWR@HTML@ref: Improved print/HTML output selection. . 461
\LWR@doindexentry: Adds support for \see, \seealso, \emph, \textbf, etc. 485
\LWR@hyperindexrefnullified: Adds support for \see, \seealso, \emph, \textbf, etc. 485
\LWR@indexitem: Accepts optional arg for **repeatindex**. 484
\dotfill: Improved print/HTML output selection. 560
\hfill: Improved print/HTML output selection. 560

\hrulefill: Improved print/HTML output selection. 560
\hyperindexref: Adds support for \see, \seealso, \emph, \textbf, etc. 485
\printindex: Fix: Extra \newpage to flush pending \index writes. . 769
General: 2018/07/07 1
*.lwarpmkconf: Added option makeindexstyle. 234
*.lwarpmkconf: Added options makeindex and xindy. 234
*.lwarpmkconf: Generated \AtBeginDocument. 234
lwarp.xdy: Requires makeindex.xdy. 275
lwarp.xdy: Supports bold, italic. 275
lwarp_html.ist: Added. 274
lwarpmk.conf: Added option makeindexstyle. 234
lwarpmk.conf: Added options makeindex and xindy. 234
lwarpmk.conf: Generated \AtBeginDocument. 234
array: Improved print/HTML output selection. 600
attachfile2: Added. 605
attachfile: Added. 604
cases: Added. 624
imakeidx: Added. 742
index: Added. 747
intopdf: Added. 749
lwarp-patch-komascript: Modified indexing. 944
lwarp-patch-memoir: Fix for \specialindex. 969
lwarp-patch-memoir: Fix for multiple indexes. 970
makeidx: Added. Moved from **lwarp** core. 769
memoir: Fix for \firsthlline, \lasthline. 442
memoir: Fix for **booktabs**. 448
pdfpages: Added. 816
pdfx: Added. 820
repeatindex: Added. 830
splitidx: Added. 857
textcomp: Improved print/HTML output selection. 869
lwarpmk: Added makeindex and xindy options. 279
lwarpmk: Added -p option for project name. 279
lwarpmk: Added optional list of names for *lwarpmk printindex* and /cmdslwarpmk htmlindex. 279
lwarpmk: Glossary generation now uses **makeglossaries**. 279
lwarpmk: *lwarpmk clean* removes all *.ind and *.idx files. 279
Added makeindex option. 191
Added xindy option. 191
Added option makeindexStyle. 190
Docs: Index, **makeindex**, **imakeidx**. 131
Docs: Misplaced \omit. 176
Fix: **memoir** and **ccaption**. 197
Improved print/HTML output selection. 559
Replaced each \csuse with \@nameuse to force error if undefined. 1
tabbing: Improved print/HTML output selection. 379
v0.59
\LWR@addbaselinemarker: Uses .eps if DVI **latex**. 494
\LWR@includegraphicsb: Fix: Expand filename. 722
Now works with .pdf and .eps filename extensions. 721
\LWR@latexmkcmd: Fix: -shell-escape with **latexmk**. . 229
\LWR@writeconf: Compilation commands now preassigned by **lwarp** instead of being computed by **lwarpmk**. 233
\[: Fix with \displaymathnormal. 503
General: 2018/09/07 1
SIunits: Added. 847
accsupp: Added. 576
amsmath: Moved from the **lwarp** core. 588
asymptote: Added. 603
axessibility: Added. 608
breqn: Added. 617
bxpapersize: Added. 618
canoniclayout: Added. 620

chemformula: Fix for \NMR. 651
draftcopy: Added. 662
epstopdf-base: Improved. 670
epstopdf: Improved. 670
fnbreak: Added. 703
nccfancyhdr: Added. 795
pdftricks: Added. 820
pst-eps: Added. 823
pstricks: Added. 823
units: Added support for **MathJax.** 912
xunicode: Added. 941
lwarpmk: Added *lwarpmk epstopdf*. 279
lwarpmk: Consolidated compiling options into printlatexcmd and HTMLlatexcmd. 279
lwarpmk: Double insead of single-dashed -shell-escape option. 279
lwarpmk: Error if lwarpmk.conf format changed. 279
lwarpmk: Warning if operating system changed. 279
Added option dvipdfmx. 192
Added option dvipdfm. 192
Added option dvips. 192
Docs: *lwarpmk epstopdf*. 97
File: lwarp_mathjax.txt: Fix: Removed chapter number from tagged non-numeric MathJax equations. 276
File: lwarp_mathjax.txt: Updated to MathJax v2.7.4. . . . 276
picture: Added an alt tag. 541
v0.60
\LWR@clearmidrules: tabular: Fix for midrules. 419
\LWR@parsenormalcolumn: tabular: Improved memory management: Not using **xstring**. 406
\LWR@tabledatasinglecolumntag: tabular: Improved memory management: Not using **xstring**. 417
\LWR@tabularendofline: Fix: Slowdown for long tables. 401
General: 2018/09/19 1
tabular: Improved memory management: Global boolean. 392, 393
tabular: Improved memory management: Not using **xstring**. 394, 791
2up: Added. 573
booklet: Added. 612
bophook: Added. 615
diagbox: Fix for par tags. 662
draftfigure: Added. 663
fancytabs: Added. 683
fullminipage: Added. 711
grid-system: Added. 729
layaureo: Added. 752
leading: Added. 752
listings: Fix for HTML entities. . . 758
listings: Fix if inside a list. . 758, 761
thumbs: Added. 878
thumb: Added. 878
twoup-gen: Added. 908
widows-and-orphans: Added. . . 920

§ 422 # Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\$ *494*
\& *304*, 6474
\(. 8863
\, ... *118*
-shell-escape (option) *100*
\@@@setcpageref 9622
\@@@setcref 9585
\@@@setcrefrange 9599
\@author *364*
\@begintheorem 6141
\@biblabel 8524
\@captype 8152
\@chapcntformat 5439
\@currentHref 8000
\@currentlabelname 7856
\@date *364*
\@dlbfloat 8118
\@donoparitem 6167
\@endtheorem 6153
\@ensuredmath 8873
\@float 8118
\@fnsymbol 5930
\@footnotetext 5090
\@include 764
\@item 6180
\@makecaption 8173
\@makefnmark 5058
\@makefntext 5057
\@maketitle 56, 5953
\@mklab 6161
\@mpfootnotetext 5092
\@nbitem 6247
\@opargbegintheorem 6147
\@rowc@lors 6795
\@rowcolors 6794
\@seccntformat 5437
\@starttoc 8292
\@textsubscript 10073
\@textsuperscript 10069
\@title *364*
\@wrglossary 8473
\@wrindex 8462
\[. 8863
\\ *561*
$ *501*
$$ *501*
~ 118
2up (package) *573*

A

a4 (package) *573*
a4wide (package) *573*
a5comb (package) *574*
abstract (environment) 6016
abstract (package) *128*, *574*
\abstractname *109*, 6015
accents 344
accsupp (package) *576*
acro (package) *577*
acronym (package) *578*
adapting
  class 174
  document 99
  package 173
\addcontentsline 8243
addlines (package) *581*
\AddSubtitlePublished 5981
adjmulticol (package) *580*
Adobe (program) *75*
ae (package) *581*
aecc (package) *581*
affiliation
  multiple authors 127
\affiliation 5833
afterpage (package) *582*
algorithm2e (package) *582*
algorithmic
  with newfloat, trivfloat 908
algorithmicx (package) *586*
algorithmx (package) *158*
align (environment) 127
align* (environment) 153
alignat (environment) 231
alignat* (environment) 257

alignment
figure or table 157
alignment tab character &, misplaced 153, 301, 388
alltt (package) 587
alt tags 148
\AmS 10333
AMSmath
split miss-numbered 149, 799
amsmath (package) 588
amsthm (package) 594
\and 363
anonchap (package) 598
anysize (package) 599
appendix (package) 128, 599
arabicfront (package) 600
array
chemformula 148
mhchem 783
array (package) 600
\arrayrulecolor 6802
\arrayrulecolornexttoken 6802
arydshln (package) 601
AsciiDoc (program) 75
AsciiDoctor (program) 75
Asciidoctor-LaTeX (program) 75
asymptote (package) 603
atbegshi (package) 603
attachfile (package) 604
attachfile2 (package) 605
\attrib 160, 375, 913
\attribution 6026
authblk (package) 128, 608
author
HTML meta tag 117, 329
multiple 127
\author 116, 363
autosec 346
axessibility (package) 608
axodraw2 (package) 609

B

babel (package) 161
\backmatter 5418
backref (package) 609
balance (package) 609
\BaseJobname 4450
BaseJobname (option) 105, 189
baseline
tabular 408
\bfseries 10058
\bibliography 8515
BibTeX
\etalchar error 129
\BibTeX 10328
bibunits (package) 610
bigdelim (package) 157, 610
bigstrut (package) 611
BlockClass (environment) 4784
\BlockClassSingle 4798
blowup (package) 612
booklet (package) 612
bookmark (package) 613
booktabs (package) 613
boolean:
CombineHigherDepths 108, 337
FileSectionNames 109, 305
FormatEPUB 163, 220
FormatWP 165, 220
HTMLDebugComments 214
LWR@amsmultline 512
LWR@copiedsidetoc 476
LWR@doingapar 321
LWR@doingcmidrule 393
LWR@doingstartpars 320
LWR@doingtbrule 393
LWR@dynamicmath 302
LWR@emptyatbang 393
LWR@exitingtabular 444
LWR@freezethisautoid 469
LWR@indisplaymathimage 494
LWR@infloatrow 720
LWR@intabularmetadata 393
LWR@isstartingequation 520
LWR@mathmacro 301
LWR@minipagefullwidth 543
LWR@minipagethispar 543
LWR@opttablecol 409
LWR@origmathjax 188
LWR@skipatbang 393
LWR@skippingmcolrowcell 393
LWR@skippingmrowcell 393
LWR@starredlongtable 436
LWR@startedrow 392
LWR@tableparcell 393
LWR@tabularcelladded 392
LWR@tabularmutemods 444
LWR@tracinglwarp 214
LWR@validtablecol 407
LWR@verbtags 376

mathjax 188
usingOSWindows 187
warpingHTML 188
warpingprint 188
WPMarkFloats 167, 220
WPMarkLOFT 168, 221
WPMarkMath 168, 221
WPMarkMinipages 167, 220
WPMarkTOC 167, 220
WPTitleHeading 168, 221
bophook (package) 615
boxedminipage (package) 615
boxedminipage2e (package) 615
\boxframe 180
breakurl (package) 616
breqn (package) 617
bugs 176
BVerbatim (environment) 222
bxpapersize (package) 618
bytefield (package) 619

C

calc (package) 206
Calibre 163
cancel (package) 619
canoniclayout (package) 620
capt-of (package) 474
caption (package) 158, 211, 474, 620
caption2 (package) 623
\caption@begin 8224
\caption@end 8224
\captionlistentry 8228
\captionof 8259
\CaptionSeparator 8172
cases (package) 624
ccaption (package) 624
center (environment) 9393
\centering 9417
\centerline 9436
changebar (package) 625
changepage (package) 625
chappg (package) 626
\chapter 5602
chapterbib (package) 626
chemfig (package) 627
chemformula
troubleshooting 148
chemformula (package) 628
chemgreek (package) 633
chemistry
Greek symbols 633
chemmacros (package) 634
chemnum (package) 654
chngpage (package) 626
cite (package) 655
class
modifying for **lwarp** 174
class (key) [Gin] 720
class:
internet 74
komascript 159
memoir 159, 160, 914
\cleardoublepage 10225
\clearpage 10225
cleveref (package) 125, 537
clrdblpg (package) 655
cmap (package) 101
color (package) 151, 655
\colorbox 73
\colorboxBlock 86, 9482
colortbl (package) 656
CombineHigherDepths (boolean) 108, 337
comment (package) 194
Computer Modern 100
\ConTeXt 10325
continue (package) 658
converting
class 174
document 99
package 173
counter:
FileDepth 108, 337
FootnoteDepth 109, 330
lofdepth 481
lotdepth 481
LWR@cellcolordepth 428
LWR@currentautosec 348
LWR@externalfilecnt 493
LWR@hdashedlines 393
LWR@hlines 392
LWR@htmlfilenumber 305
LWR@latestautopage 471
LWR@lateximagedepth 517
LWR@lateximagenumber 517
LWR@LIpage 517
LWR@midrulecounter 419
LWR@minipagedepth 542
LWR@nextautoid 471
LWR@nextautopage 471

LWR@nextequation 505
LWR@prevFileDepth 347
LWR@previousautopagelabel . . . 339
LWR@spandepth 320
LWR@startingequation 520
LWR@tablecolspecindex 394
LWR@tablecolspecwidth 394
LWR@tableLaTeXcolindex 395
LWR@tabletotalLaTeXcols 395
LWR@tabletotalLaTeXcolsnext . . 395
LWR@tabularDepth 393
LWR@tabularpardepth 394
LWR@thisautoid 469
LWR@thisautoidWP 469
SideTOCDepth 108, 479
tocdepth 108
\cpagerefFor 9621
crop (package) 658
cross-references
missing or incorrect 178
CSS
file selection 112
lwarp.css 113
per HTML page 112
project-specific changes 113
\CSSFilename 109, 112, 5013
\CustomizeMathJax 5284, 5309
cuted (package) 658
cutwin (package) 659

D

danger icon 183
\date 116
dblfloatfix (package) 660
dblfnote (package) 660
dcolumn (package) 660
debugging 176
HTML debug comments 214
\DeclareGraphicsExtensions 2
Deja Vu 100
description
HTML meta tag 116, 329
description (environment) 6322
diagbox (package) 661
display math
complicated objects 148
\displaymathnormal . 148, 493, 510, 9055
\displaymathother . . 148, 493, 510, 9066
document
convert existing 99
\dotfill 10146
\doublerulesepcolor 6804
\doublerulesepcolornexttoken . . . 6804
draftcopy (package) 662
draftfigure (package) 663
draftwatermark (package) 663
dvipdfm (option) 102, 192
dvipdfmx (option) 102, 192
dvips (option) 102, 192
dynamic math expressions 302

E

easy-todo (package) 664
ebook (package) 665
ellipsis (package) 665
\emph 9932
emptypage (package) 665
\end@dlbfloat 8135
\end@float 8135
endfloat (package) 666
endheads (package) 666
endnotes (package) 129, 667
\enlargethispage 10224
\enskip 561, 10170
enumerate (environment) 6303
enumerate (package) 668
enumitem (package) 668
environ (package) 207
environment:
fminipage 122
lateximage 516
picture 541
titlepage 116
titlingpage 116
warpall 114
warpHTML 111, 114
warpprint 111, 114
environments:
abstract 6016
align 127
align* 153
alignat 231
alignat* 257
BlockClass 4784
BVerbatim 222
center 9393
description 6322
enumerate 6303
eqnarray 9115
equation 9033

equation* 9041
fcolorminipage 161, 9523
flalign 179
flalign* 205
flushleft 9409
flushright 9401
fminipage 9870
gather 75
gather* 101
itemize 6297
lateximage 9263, 9388
list 6267
longtable 2
LWR@BlockClassWP 4821
LWR@blocktextcurrentfont . . 10050
LWR@displaymathnormal 8903
LWR@displaymathother 8908
LWR@equationother 8922
LWR@figcaption 8187
LWR@nestspan 4648
LWR@tabular 7743
math 8902
minipage 9700
multline 20
multline* 47
picture 9681
quote 6042
tabbing 6126
thebibliography 8525
theindex 8438
titlepage 5850
titlingpage 14
verbatim 6116
verse 2, 6057
warpall 208
warpHTML 101, 209
warpprint 209
epigraph (package) *669*
EPS images
converting 97
using 150, 531
epstopdf (package) *152*, *670*
epstopdf (program) *150*, *532*
epstopdf-base (package) *670*
EPUB
conversion software 163
HTML conversion settings . . . 163, 220
eqnarray (environment) 9115
equation
miss-numbered 149, 799
with Ti*k*z 148
equation (environment) 9033
equation* (environment) 9041
errata (package) *671*
error
Improper \prevdepth. 129, 177
Label(s) may have changed. 176
Leaders not followed by proper glue. 176
Missing $ inserted. 176
No Room for a new \write. 176
Options clash, memoir 160
error messages 176
eso-pic (package) *673*
\etalchar *129*
etoolbox (package) *186*
everyhook (package) *205*
everypage (package) *674*
everyshi (package) *674*
expl3 (package) *205*
export
to word processor 165
extramarks (package) *674*

F

fancybox (package) *123*, *675*
fancyhdr (package) *681*
fancyheadings (package) *681*
fancyref (package) *682*
fancytabs (package) *683*
fancyvrb (package) *683*
FAQ 176
fbox
around a minipage 122
\fbox *122*, 9845
\fboxBlock *122*, 9853
\fcolorbox 101, 9484
\fcolorboxBlock 124, 9521
fcolorminipage (environment) 161, 9523
figcaps (package) *690*
figsize (package) *690*
figure
text alignment 157
file
hashed name 495, 522
inaccessible 108
underscore in name 106, 120
unique names 109
file:
glyphtounicode.tex *101*
lwarp.css *113*, *234*

lwarp.ist 141, 274
lwarp.xdy 142, 275
lwarp_baseline_marker.eps . . . 494
lwarp_baseline_marker.png . . . 494
lwarp_formal.css 269
lwarp_mathjax.txt 276
lwarp_one_limage.cmd 275
lwarp_sagebrush.css 264
lwarp_tutorial.txt 85
lwarpmk.conf 234
lwarpmk.lua 175
project.css 113
project.lwarpmkconf 234
project_html.tex 227
sample_project.css 113, 273
tutorial.tex 85
filecontents (package) 206
FileDepth (counter) 108, 337
FileSectionNames (boolean) . . . 109, 305
fix2col (package) 691
fixme (package) 162, 691
fixmetodonotes (package) 692
flafter (package) 693
\flagverse 914
flalign (environment) 179
flalign* (environment) 205
Flare (program) 75
float
text alignment 157
float (package) 158, 693
floatflt (package) 695
floatpag (package) 696
floatrow (package) 158, 696
fltrace (package) 702
\flushbottom 4566
flushend (package) 702
flushleft (environment) 9409
flushright (environment) 9401
fminipage (environment) 122, 9870
fnbreak (package) 703
fncychap (package) 703
fnlineno (package) 704
fnpos (package) 704
font
bitmapped 100
Computer Modern 100
Deja Vu 100
ligatures 101
missing symbols 119
packages 101
selection 100
size — lateximage 145, 490, 517
size — math, SVG 145, 490, 517
size — xfrac 934
fontenc (package) 101, 704
fontspec (package) 101, 705
footmisc (package) 705
footnote (package) 706
FootnoteDepth (counter) 109, 330
footnotehyper (package) 707
footnotes 329
in math 126
in sectioning commands 126
misnumbered 126
verbatim 126
footnpag (package) 708
for
HTML & PRINT 183
HTML output 183
PRINT output 183
\ForceHTMLPage 127, 5393
\ForceHTMLTOC 128, 5399
forest (package) 708
FormatEPUB (boolean) 163, 220
FormatWP (boolean) 165, 220
\framebox 9823
framed (package) 709
framed objects 122
FrameMaker (program) 75
Frequently Asked Questions 176
\frontmatter 5415
ftnright (package) 711
fullminipage (package) 711
fullpage (package) 712
fullwidth (package) 712
\fup 10077
\fussy 4568
fwlw (package) 712

G

gather (environment) 75
gather* (environment) 101
GELLMU (program) 74
generator
HTML meta tag 357
geometry (package) 204, 713
gettitlestring (package) 205
GIF images 150, 532
[Gin]:
class (key) 720

GladTeX (program) 74
glossaries (package) 129, 713
glossary
  language 129
  options 129
  processing 96
glossary (package) 716
GlossaryCmd (option) . . 105, 129, 192, 713
glyphtounicode.tex (file) 101
graphics
  formats 150, 531
graphics (package) 150, 716
graphicx (package) 150, 728
Greek
  chemistry symbols 633
grffile (package) 151, 729
grid (package) 729
grid-system (package) 729
gridset (package) 730

H

hang (package) 730
hanging (package) 732
hash
  SVG image filename 495, 522
heading, word processor 168
Hevea (program) 74
\hfill 10142
\HomeHTMLFilename 4452
HomeHTMLFilename (option) . 102, 106, 191
horizontal space 118
  between minipages 561
\href 8029
\hrulefill 10144
\hskip 119
\hspace . . . 118, 561, 10178, 10208, 10214
htlatex (program) 74
HTML
  alt tags 148
  appearing instead of images . . 91, 179
  commands for a successful conversion 117
  conversion settings 106
    debug comments 214
    EPUB 163, 220
    word processor 165, 220
  filename generation 111
  headings 182
  meta tag
    author 117, 329
    description 116, 329
    generator 357
    Title 328
    title 117, 328
    viewport 357
  selecting print/HTML definitions . . 215
  tabular column conversion 408
\HTMLAuthor 109, 117, 5039
HTMLDebugComments (boolean) 214
\HTMLDescription 110, 116, 5044
\HTMLentity 4433
\HTMLFilename 4451
HTMLFilename (option) 102, 106, 191
\HTMLFirstPageTop 110, 5001
htmlglossary (option) [lwarpmk] 129, 713
HTMLIndexCmd (option) 104, 191
\HTMLLanguage 109, 5676
\HTMLleftmargini (length) 160, 375, 376, 913
\HTMLPageBottom 110, 364, 5007
\HTMLPageTop 110, 5004
\HTMLTitle 109, 117, 5034
\HTMLunicode 4441
\HTMLvleftskip (length) 160, 375, 376, 913
hypcap (package) 733
hypdestopt (package) 733
\hyperindexref 8497
hypernat (package) 733
hyperref
  and xindy 131
  incorrect links 178
hyperref (package) 125, 462, 734
\hypertoc 8385
\hypertocfloat 8405
hyperxmp (package) 739
hyphenat (package) 740

I

icon
  warning 183
idxlayout (package) 741
\if@titlepage 5828
ifoddpage (package) 741
ifplatform (package) 187
images
  appearing as HTMLcode 91, 179
  format priorities 150, 532
  GIF 150, 532
  graphicx package 531
  hashed filenames 495, 522

in strange places 178
PDF or EPS
converting 97, 150, 532
using 150, 531
PNG and JPG 150, 532
processing 279
imakeidx (package) *742*
Improper \prevdepth. 129, 177
\include 764
\includegraphics
using 150, 531
\includegraphics 245
indentfirst (package) *747*
InDesign (program) *75*
index
custom **makeindex** style file 141
custom **xindy** style file 142
custom display styles 143
empty 143
letter headings 742
missing entries 131
placement and TOC options . . 143, 893
processing 89, 90, 131
see, seealso, ranges 131
source code 131
table of contents 143, 893
UTF-8 101
xindy and hyperref 131
index (package) *747*
\InlineClass 4809
inputenc (package) *101*, *749*
inputenx (package) *101*, *749*
internet (class) *74*
intopdf (package) *749*
item
empty 119, 380
itemize (environment) 6297
\itshape 10063

J

JavaScript
MathJax 144, 489
JPG images 150, 532

K

\kern *119*
key:
[Gin]:
class *720*
keyfloat (package) *159*, *750*
komascript (class) *159*
kvoptions (package) *188*

L

\l@chapter 8422
\l@figure 8429
\l@paragraph 8427
\l@part 8421
\l@section 8424
\l@subparagraph 8428
\l@subsection 8425
\l@subsubsection 8426
\l@table 8430
label
in HTML 360
math environment 512
valid characters 125
Label(s) may have changed 176
language
glossary 129
language HTML metadata 356
\LaTeX 10301
LaTeX2HTML (program) *74*
\LaTeXe 10301
lateximage (environment) *516*, 9263, 9388
\LateximageFontScale 9185
\LateximageFontSizeName 9184
lateximages
font size 145, 490, 517
processing 279
latexmk (option) *102*, *106*, *192*
LatexmkIndexCmd (option) *104*, *191*
LaTeXML (program) *74*
layaureo (package) *752*
layout (package) *752*
Leaders not followed by proper glue. . . 176
leading (package) *752*
\leftline 9435
length:
\HTMLleftmargini . *160*, *375*, *376*, *913*
\HTMLvleftskip . . . *160*, *375*, *376*, *913*
\LWR@cmidrulewidth *419*
\LWR@heavyrulewidth *419*
\LWR@lightrulewidth *419*
\LWR@minipageheight *542*
\LWR@minipagewidth *542*
\LWR@tempheight *562*
\LWR@tempraise *562*
\LWR@tempwidth *562*
\LWR@thiscmidrulewidth *419*

\VerbatimHTMLWidth *376*
\vleftmargini *160*, *375*, *913*
\vleftskip *160*, *375*, *913*, *914*
letltxmacro (package) *187*
letterspace (package) *753*
lettrine (package) *753*
LibreOffice
conversion recommendations 168
import into 165
section headings 168
LibreOffice (program) *75*
ligatures 101, 204
line numbers 183
\linebreak 10218
lineno (package) *754*
\LinkHome 4505
Linux (program) *113*, *187*
lips (package) *757*
list
empty item 119, 380
filename 120
list (environment) 6267
listings (package) *757*
\listof 8346
\listoffigures 8320
\listoftables 8333
lmodern (package) *101*
lofdepth (counter) *481*
longtable (environment) 2
longtable (package) *156*, *762*
lotdepth (counter) *481*
lscape (package) *764*
ltablex (package) *764*
ltcaption (package) *765*
ltxgrid (package) *765*
ltxtable (package) *765*
lua-check-hyphen (package) *766*
luacolor (package) *766*
LuaLaTeX
detection 184
file & section names 344
\LuaLaTeX 10319
LuaLaTeX (program) [requirement] . . . *78*
\LuaTeX 10319
luatodonotes (package) *162*, *766*
lwarp
loading 102
options 102
lwarp (package) *102*
lwarp-patch-komascript (package) . *944*
lwarp-patch-memoir (package) *946*
lwarp.css (file) *113*, *234*
lwarp.ist
customizing 141
lwarp.ist (file) *141*, *274*
lwarp.xdy
customizing 142
lwarp.xdy (file) *142*, *275*
lwarp_baseline_marker.eps (file) . . *494*
lwarp_baseline_marker.png (file) . . *494*
lwarp_formal.css (file) *269*
lwarp_mathjax.txt (file) *276*
lwarp_one_limage.cmd (file) *275*
lwarp_sagebrush.css (file) *264*
lwarp_tutorial.txt (file) *85*
lwarpmk
customizing 175
[lwarpmk]:
htmlglossary (option) *129*, *713*
printglossary (option) *129*, *713*
lwarpmk (option) *105*, *190*, *278*
lwarpmk (program) *175*, *279*
lwarpmk epstopdf (program) . . *150*, *532*
lwarpmk pdftosvg (program) . . *150*, *532*
lwarpmk.conf (file) *234*
lwarpmk.lua (file) *175*
\LWR@addbaselinemarker 8626
\LWR@addcdashline 7073
\LWR@addcmidruletrim 7039
\LWR@addcmidrulewidth 7070
\LWR@addcompilecmd 1151
\LWR@addformatwpalignment 7089
\LWR@addleftmostbartag 6874
\LWR@addmathjax 8971
\LWR@addmulticolvertrulecolor . . 7348
\LWR@addrulewidth 7044
\LWR@addtabularcellcolor 7217
\LWR@addtabularhrulecolor 7117
\LWR@addtabularrowcolor 7102
\LWR@addtabularrulecolors 7163
\LWR@afterendverbatim 6099
\LWR@amsmathbody 9250
\LWR@amsmathbodynumbered 9256
LWR@amsmultline (boolean) *512*
\LWR@atbeginverbatim 6080
\LWR@backgroundcolor 36
\LWR@beginhideamsmath 9100
\LWR@blackborderpadding 9841
LWR@BlockClassWP (environment) . . 4821

LWR@blocktextcurrentfont (environment) 10050
\LWR@botnavigation 4511
LWR@cdashlines 419
LWR@cellcolordepth (counter) 428
\LWR@cellHTMLcolor 6799
\LWR@clearmidrules 6954
\LWR@closeparagraph 4907
\LWR@closeprevious 4538
\LWR@closetabledatacell 6434
\LWR@cmidrulewidth (length) 419
LWR@colafterspec 395
LWR@colatspec 395
LWR@colbangspec 395
LWR@colbarspec 395
LWR@colbeforespec 395
\LWR@columnHTMLcolor 6797
\LWR@columnspeclookahead 6552
\LWR@compilecmd 1148
\LWR@convertto 66
LWR@copiedsidetoc (boolean) 476
\LWR@copyfile 799
\LWR@createautosec 5424
LWR@currentautosec (counter) 348
\LWR@currentcss 5012
\LWR@currenttextcolor 10088
\LWR@customizedMathJax 5283
\LWR@customizeMathJax 5289
\LWR@descitem 6309
LWR@displaymathnormal (environment) 8903
LWR@displaymathother (environment) 8908
\LWR@docdashline 7003
\LWR@docmidrule 6987
\LWR@doequation 8984
\LWR@doindexentry 8481
LWR@doingapar (boolean) 321
LWR@doingcmidrule (boolean) 393
LWR@doingstartpars (boolean) 320
LWR@doingtbrule (boolean) 393
\LWR@domulticolumn 7386
\LWR@doubledollar 8818
LWR@dynamicmath (boolean) 302
\LWR@earlyloadnever 250
LWR@emptyatbang (boolean) 393
\LWR@endfloatalignment 8168
\LWR@endhideamsmath 9108
\LWR@endofline 10153
\LWR@ensuredoingapar 4883
LWR@equationother (environment) . 8922
\LWR@equationtag 9236
LWR@exitingtabular (boolean) 444
\LWR@expandableformatted 861
\LWR@expandableformattedenv 886
LWR@externalfilecnt (counter) 493
LWR@figcaption (environment) 8187
\LWR@filenamenoblanks 5195
\LWR@filestart 5683
\LWR@findcurrenttextcolor 10091
\LWR@findword 675
\LWR@floatalignment 8154
\LWR@floatalignmentname 8153
\LWR@floatbegin 8086
\LWR@floatend 8120
\LWR@floatstyle 2
\LWR@footnotetext 5061
\LWR@forceminwidth 9830
\LWR@forcenewpage 4555
\LWR@formatted 846
\LWR@formattedenv 876
\LWR@formatting 845
LWR@freezethisautoid (boolean) . . . 469
\LWR@futurenonspacelet 6381
\LWR@FVstyle 56
\LWR@getexparray 4385
\LWR@getmynexttoken 6388
LWR@hdashedlines (counter) 393
\LWR@heavyrulewidth (length) 419
\LWR@hidelatexequation 8957
LWR@hlines (counter) 392
\LWR@HTML@caption@begin 8203
\LWR@HTML@caption@end 8215
\LWR@HTML@Gin@setfile 101
\LWR@HTML@ref 7971
\LWR@htmlblockcomment 4734
\LWR@htmlblocktag 4736
\LWR@HTMLcline 7728
\LWR@htmlclosecomment 4706
\LWR@htmlcomment 4727
\LWR@htmldivclass 4761
\LWR@htmldivclassend 4764
\LWR@htmlelement 4774
\LWR@htmlelementclass 4748
\LWR@htmlelementclassend 4753
\LWR@htmlelementclassline 4767
\LWR@htmlelementend 4777
LWR@htmlfilenumber (counter) 305
\LWR@htmlfileref 7883
\LWR@HTMLhline 7711
\LWR@HTMLlatexcmd 1180

\LWR@htmlmathlabel 9085
\LWR@htmlmulticolumn 7435
\LWR@htmlopencomment 4706
\LWR@htmlrefsectionfilename 4495
\LWR@HTMLsanitize 9188
\LWR@HTMLsanitizeexpand 9205
\LWR@htmlsectionfilename 4462
\LWR@htmlspan 4683
\LWR@htmlspanclass 4691
\LWR@htmltag 4701
\LWR@htmltagc 4637
\LWR@HTMLtextstyle 9927
\LWR@hyperindexrefnullified 8486
\LWR@imagesizebox 100
\LWR@includegraphicsb 118
\LWR@indexitem 8450
\LWR@indexnameref 8480
\LWR@indexsubitem 8454
\LWR@indexsubsubitem 8458
LWR@indisplaymathimage (boolean) . . *494*
LWR@infloatrow (boolean) *720*
\LWR@instertatbangcols 6426
LWR@intabularmetadata (boolean) . . *393*
LWR@isstartingequation (boolean) . . *520*
\LWR@itemizeitem 6290
LWR@latestautopage (counter) *471*
LWR@lateximagedepth (counter) *517*
\LWR@lateximagedepthref 7886
LWR@lateximagenumber (counter) . . . *517*
\LWR@lateximagenumberref 7889
\LWR@lateximagesfile 761
\LWR@latexmkcmd 1166
\LWR@latexmkdvipdfm 1171
\LWR@latexmkvar 1160
\LWR@lightrulewidth (length) *419*
LWR@LIpage (counter) *517*
\LWR@listitem 6252
\LWR@loadafter 219
\LWR@loadbefore 233
\LWR@loadnever 242
\LWR@longtabledatacaptiontag . . . 7446
\LWR@lookforpackagename 678
\LWR@lwarpconfversion 1126
\LWR@LwarpEnd 5804, 10355
\LWR@LwarpStart 5733, 10355
\LWR@maketitlesetup 33, 5922
LWR@mathmacro (boolean) *301*
\LWR@maybenewtablerow 6806
\LWR@maybeprintpendingfootnotes 5135
LWR@midrulecounter (counter) *419*
LWR@midrules *418*
LWR@minipagedepth (counter) *542*
LWR@minipagefullwidth (boolean) . . *543*
\LWR@minipageheight (length) *542*
\LWR@minipagestartpars 10158
\LWR@minipagestoppars 10161
LWR@minipagethispar (boolean) *543*
\LWR@minipagewidth (length) *542*
\LWR@multicolother 7289
\LWR@multicolpartext 7283
\LWR@multicolskip 7295
\LWR@multirowborder 3
\LWR@mynexttoken 6380
\LWR@myshorttoc 8269
\LWR@nameref 7880
LWR@nestspan (environment) 4648
\LWR@new@label 7920
\LWR@newautoidanchor 8141
\LWR@newautopagelabel 5275
\LWR@newhtmlfile 5312
LWR@nextautoid (counter) *471*
LWR@nextautopage (counter) *471*
LWR@nextequation (counter) *505*
\LWR@notmemoirloadafter 230
\LWR@nullfonts 9997
\LWR@nullifyNoAutoSpacing 7730
\LWR@nulllistfills 6259
\LWR@opacity 99
\LWR@openparagraph 4889
\LWR@opquote 1133
\LWR@opsequence 1138
LWR@opttablecol (boolean) *409*
\LWR@origcolspec 6420
\LWR@originname 259
\LWR@originnames 266
LWR@origmathjax (boolean) *188*
\LWR@overline 10083
\LWR@parseaftercolumn 6622
\LWR@parseatcolumn 6562
\LWR@parsebangcolumn 6588
\LWR@parsebarcolumn 6633
\LWR@parsebeforecolumn 6611
\LWR@parsecoloncolumn 6658
\LWR@parseDcolumn 6704
\LWR@parsedrequirepackagenames . . 674
\LWR@parsenormalcolumn 6683
\LWR@parsepcolumn 6695
\LWR@parsesemicoloncolumn 6679
\LWR@parsetablecols 6710
\LWR@parsewcolumn 6699

\LWR@patcherror 67
\LWR@patchlists 6328
LWR@prevFileDepth (counter) *347*
LWR@previousautopagelabel (counter) *339*
\LWR@printatbang 6849
\LWR@printbartag 6839
\LWR@printcloselist 6159
\LWR@printlatexcmd 1180
\LWR@printlength 656
\LWR@printmccoldata 7298
\LWR@printmccoltype 7241
\LWR@printopenlist 6160
\LWR@printpendingfootnotes 5125
\LWR@printpendingmpfootnotes . . . 5144
\LWR@printthetitle 5874
\LWR@providelength 63
\LWR@ProvidesPackageDrop 742
\LWR@ProvidesPackagePass 731
\LWR@pushoneclose 5427
\LWR@quickfile 758
\LWR@ref@ignorestar 7979
\LWR@remembertag 9240
\LWR@replaceEPSSVG 107
\LWR@requesttoc 5797
\LWR@requirepackagenames 673
\LWR@restoreorigaccents 1093
\LWR@restoreorigformatting 8544
\LWR@restoreoriglists 6356
\LWR@rotboxorigin 258
\LWR@rotstyle 91
\LWR@rowHTMLcolor 6798
\LWR@ruleHTMLcolor 6800
\LWR@sanitize 4423
\LWR@sanitized 4422
\LWR@scalestyle 94
\LWR@section 5441
\LWR@sectionnumber 5421
\LWR@select@html@hspace 10178
\LWR@select@html@nohspace 10208
\LWR@select@print@hspace 10211
\LWR@setexparray 4379
\LWR@setlatestname 7858
\LWR@setOSWindows 79
\LWR@shellescapecmd 1143
\LWR@sidetoc 8362
\LWR@singledollar 8842
LWR@skipatbang (boolean) *393*
LWR@skippingmcolrowcell (boolean) . *393*
LWR@skippingmrowcell (boolean) . . . *393*
LWR@spandepth (counter) *320*
\LWR@splabel 7892
LWR@starredlongtable (boolean) . . . *436*
LWR@startedrow (boolean) *392*
LWR@startingequation (counter) . . . *520*
\LWR@startingequationtag 9235
\LWR@startnewdepth 5431
\LWR@startpars 4956
\LWR@startref 7933
\LWR@stoppars 4974
\LWR@stripperiod 7857
\LWR@strresult 6418
\LWR@subaddcmidruletrim 7026
\LWR@subaddtabularcellcolor 7211
\LWR@subcdashline 6990
\LWR@subcmidrule 6972
\LWR@subhtmlelementclass 4741
\LWR@subhyperref 8005
\LWR@subhyperrefclass 8018
\LWR@subhyperreftext 8013
\LWR@subinlineimage 8079
\LWR@sublabel 7897
\LWR@subnewref 7965
\LWR@subsingledollar 8662
\LWR@subtableofcontents 8282
\LWR@subtabularhtmlcolumns 7516
\LWR@syncmathjax 8937
LWR@tablecolspec *394*
LWR@tablecolspecindex (counter) . . . *394*
LWR@tablecolspecwidth (counter) . . . *394*
\LWR@tabledatacolumntag 7607
\LWR@tabledatasinglecolumntag . . 6888
LWR@tableLaTeXcolindex (counter) . . *395*
LWR@tableparcell (boolean) *393*
LWR@tabletotalLaTeXcols (counter) . *395*
LWR@tabletotalLaTeXcolsnext (counter) *395*
LWR@tabular (environment) 7743
LWR@tabularcelladded (boolean) . . . *392*
LWR@tabularDepth (counter) *393*
\LWR@tabularendofline 6532
\LWR@tabularfinishrow 6486
\LWR@tabularhtmlcolumns 7526
\LWR@tabularleftedge 6879
LWR@tabularmutemods (boolean) *444*
LWR@tabularpardepth (counter) *394*
\LWR@tdaddstyle 7014
\LWR@tdendstyles 7020
\LWR@tdstartstyles 7013
\LWR@tempcolor 10089
\LWR@tempcolortwo 10089

\LWR@tempheight (length) *562*
\LWR@tempraise (length) *562*
\LWR@tempwidth (length) *562*
\LWR@textcurrentcolor 19, 10094
\LWR@textcurrentfont 10042
LWR@thisautoid (counter) *469*
LWR@thisautoidWP (counter) *469*
\LWR@thiscmidrulewidth (length) . . . *419*
\LWR@thiscolspec 6887
\LWR@thisfilename 5193
\LWR@thisnewfilename 5194
\LWR@titlingmaketitle 83, 5973
\LWR@topnavigation 4508
\LWR@traceinfo 827
LWR@tracinglwarp (boolean) *214*
LWR@trimlrules *419*
LWR@trimrrules *419*
\LWR@unknownengine 1155
LWR@validtablecol (boolean) *407*
LWR@verbtags (boolean) *376*
\LWR@vspace 10215
\LWR@WPcell 7085
\LWR@writeconf 1270
\LWR@xcolorrowHTMLcolor 6796
\LWRPrintStack 4516
\LWRsetnextfloat 8181
\LyX 10336

M

Mac OS (program) *113*, *187*
Madcap (program) *75*
magaz (package) *769*
\mainmatter 5411
\makebox 9801
makeidx (package) . *143*, *144*, *769*, *893*, *894*
\MakeIndex 10328
makeindex
  customizing 141
makeindex (option) *102*, *106*, *191*
makeindex (program) *141*
makeindexStyle (option) . . . *102*, *142*, *190*
\makelabel *381*
\maketitle 47, *116*, 5934
margin
  numbers 183
  tags 183
marginfit (package) *770*
marginfix (package) *770*
marginnote (package) *770*
\marginpar *120*, *335*, 5156
\marginparBlock . . . *120*, *335*, 5167, 5184
\markboth 4563
\markright 4564
markup languages 75
math
  \displayingmath 148
  \displayingmore 148
  tabbing 153
  mathjax option 189
  mathsvg option 189
  chemformula 148
  mhchem 783
  Ti*k*z 148
  appearing as HTMLcode 91, 179
  display with complicated objects . . 148
  dynamic 302
  font size — SVG 145, 490, 517
  footnotes 126
  images incorrect 302
  labels — valid characters 125
  MathJax custom functions . . . 146, 491
  MathJax summary 146, 490
  SVG summary 145, 490
  word processor conversion 168
math (environment) 8902
\mathimagename *110*, 8613
MathJax
  mathjax option 189
  custom functions 146, 491
  errors 148
  subequations 147, 491
  summary 146, 490
  tagged equations 146, 491
MathJax (program) *146*, *490*, *491*
MathJax (program) [requirement] *78*
mathjax (boolean) *188*
mathjax (option) *102*, *106*, *189*
mathsvg (option) *102*, *106*, *189*
\mbox 9798
mcaption (package) *771*
\mcolrowcell 7682
MD5 hash
  SVG image filename 495, 522
mdframed (package) *124*, *771*
\mdseries 10057
memhfixc (package) *782*
memoir
  options clash 160
  verse 914
memoir (class) *159*, *160*, *914*

meta tag, HTML
author . . . 117, 329
description . . . 116, 329
generator . . . 357
Title . . . 328
title . . . 117, 328
viewport . . . 357
metalogo (package) . . . *782*
mhchem
troubleshooting . . . 783
mhchem (package) . . . *782*
microtype (package) . . . *204*, *785*
midfloat (package) . . . *786*
midpage (package) . . . *786*
\MiKTeX . . . 10335
minipage
framed . . . 122
horizontal space between . . . 561
minipage (environment) . . . 9700
\minipagefullwidth . . . 9692
misplaced \noalign . . . 155, 390
misplaced alignment tab character & . . . 153, 301, 388
Missing $ inserted. . . . 176
missing sections . . . 108
modifying
class . . . 174
document . . . 99
package . . . 173
morefloats (package) . . . *786*
moreverb (package) . . . *787*
morewrites
No room for a new \write. . . . 176
morewrites (package) . . . *788*
mparhack (package) . . . *788*
\mrowcell . . . 7679
MS-Windows (program) . . . *114*, *187*
multicol (package) . . . *789*
multicolumn
with multirow . . . 792
\multicolumnrow . . . 39, 7549
multirow
with multicolumn . . . 792
\multirow . . . *791*
multirow (package) . . . *790*
multitoc (package) . . . *793*
multline (environment) . . . 20
multline* (environment) . . . 47

N

\Nameref . . . 7997
\nameref . . . 7988
nameref (package) . . . *794*
natbib (package) . . . *794*
nccfancyhdr (package) . . . *795*
needspace (package) . . . *796*
newclude (package) . . . *161*, *796*
newfloat
with trivfloat, algorithmic . . . 908
\newfloat . . . 4
newfloat (package) . . . *206*
\newline . . . 10151
\newpage . . . 10148
\newtheorem . . . *379*
newtxmath (package) . . . *149*
newunicodechar (package) . . . *101*, *796*
nextpage (package) . . . *797*
nicefrac (package) . . . *149*, *797*
No room for a new \write. . . . 176
\noalign . . . 7700
\nolinebreak . . . 10219
\nolinkurl . . . 8045
nonfloat (package) . . . *798*
nonumonpart (package) . . . *798*
\nopagebreak . . . 10223
\nopagecolor . . . 72
nopageno (package) . . . *798*
\normalfont . . . 10065
\normalmarginpar . . . 5181
nowidow (package) . . . *799*
ntheorem (package) . . . *149*, *799*
\numberline . . . 8380
numbers
left margin . . . 183
numindex (option) [tocbibind] . . . *144*, *894*

O

OpenOffice (program) . . . *75*
option:
-shell-escape . . . *100*
[lwarpmk]:
htmlglossary . . . *129*, *713*
printglossary . . . *129*, *713*
[tocbibind]:
numindex . . . *144*, *894*
[tocloft]:
titles . . . *128*
BaseJobname . . . *105*, *189*
dvipdfm . . . *102*, *192*

dvipdfmx 102, 192
dvips 102, 192
GlossaryCmd 105, 129, 192, 713
HomeHTMLFilename 102, 106, 191
HTMLFilename 102, 106, 191
HTMLIndexCmd 104, 191
latexmk 102, 106, 192
LatexmkIndexCmd 104, 191
lwarpmk 105, 190, 278
makeindex 102, 106, 191
makeindexStyle 102, 142, 190
mathjax 102, 106, 189
mathsvg 102, 106, 189
OSWindows 105, 114, 187, 190
pdftotextEnc 105, 190
PrintIndexCmd 102, 191
warpHTML 105, 189
warpprint 105, 189
xindy 102, 106, 191
xindyCodepage 102, 190
xindyLanguage 102, 190
xindyStyle 102, 142, 190
options clash, memoir 160
\OSPathSymbol 78
OSWindows (option) 105, 114, 187, 190
overpic (package) 153, 812

P

package
modifying for **lwarp** 173
required 204
package:
2up 573
a4 573
a4wide 573
a5comb 574
abstract 128, 574
accsupp 576
acro 577
acronym 578
addlines 581
adjmulticol 580
ae 581
aecc 581
afterpage 582
algorithm2e 582
algorithmicx 586
algorithmx 158
alltt 587
amsmath 588
amsthm 594
anonchap 598
anysize 599
appendix 128, 599
arabicfront 600
array 600
arydshln 601
asymptote 603
atbegshi 603
attachfile 604
attachfile2 605
authblk 128, 608
axessibility 608
axodraw2 609
babel 161
backref 609
balance 609
bibunits 610
bigdelim 157, 610
bigstrut 611
blowup 612
booklet 612
bookmark 613
booktabs 613
bophook 615
boxedminipage 615
boxedminipage2e 615
breakurl 616
breqn 617
bxpapersize 618
bytefield 619
calc 206
cancel 619
canoniclayout 620
capt-of 474
caption 158, 211, 474, 620
caption2 623
cases 624
ccaption 624
changebar 625
changepage 625
chappg 626
chapterbib 626
chemfig 627
chemformula 628
chemgreek 633
chemmacros 634
chemnum 654
chngpage 626
cite 655

cleveref 125, 537
clrdblpg 655
cmap 101
color 151, 655
colortbl 656
comment 194
continue 658
crop 658
cuted 658
cutwin 659
dblfloatfix 660
dblfnote 660
dcolumn 660
diagbox 661
draftcopy 662
draftfigure 663
draftwatermark 663
easy-todo 664
ebook 665
ellipsis 665
emptypage 665
endfloat 666
endheads 666
endnotes 129, 667
enumerate 668
enumitem 668
environ 207
epigraph 669
epstopdf 152, 670
epstopdf-base 670
errata 671
eso-pic 673
etoolbox 186
everyhook 205
everypage 674
everyshi 674
expl3 205
extramarks 674
fancybox 123, 675
fancyhdr 681
fancyheadings 681
fancyref 682
fancytabs 683
fancyvrb 683
figcaps 690
figsize 690
filecontents 206
fix2col 691
fixme 162, 691
fixmetodonotes 692
flafter 693
float 158, 693
floatflt 695
floatpag 696
floatrow 158, 696
fltrace 702
flushend 702
fnbreak 703
fncychap 703
fnlineno 704
fnpos 704
fontenc 101, 704
fontspec 101, 705
footmisc 705
footnote 706
footnotehyper 707
footnpag 708
forest 708
framed 709
ftnright 711
fullminipage 711
fullpage 712
fullwidth 712
fwlw 712
geometry 204, 713
gettitlestring 205
glossaries 129, 713
glossary 716
graphics 150, 716
graphicx 150, 728
grffile 151, 729
grid 729
grid-system 729
gridset 730
hang 730
hanging 732
hypcap 733
hypdestopt 733
hypernat 733
hyperref 125, 462, 734
hyperxmp 739
hyphenat 740
idxlayout 741
ifoddpage 741
ifplatform 187
imakeidx 742
indentfirst 747
index 747
inputenc 101, 749
inputenx 101, 749

intopdf 749
keyfloat 159, 750
kvoptions 188
layaureo 752
layout 752
leading 752
letltxmacro 187
letterspace 753
lettrine 753
lineno 754
lips 757
listings 757
lmodern 101
longtable 156, 762
lscape 764
ltablex 764
ltcaption 765
ltxgrid 765
ltxtable 765
lua-check-hyphen 766
luacolor 766
luatodonotes 162, 766
lwarp 102
lwarp-patch-komascript 944
lwarp-patch-memoir 946
magaz 769
makeidx 143, 144, 769, 893, 894
marginfit 770
marginfix 770
marginnote 770
mcaption 771
mdframed 124, 771
memhfixc 782
metalogo 782
mhchem 782
microtype 204, 785
midfloat 786
midpage 786
morefloats 786
moreverb 787
morewrites 788
mparhack 788
multicol 789
multirow 790
multitoc 793
nameref 794
natbib 794
nccfancyhdr 795
needspace 796
newclude 161, 796
newfloat 206
newtxmath 149
newunicodechar 101, 796
nextpage 797
nicefrac 149, 797
nonfloat 798
nonumonpart 798
nopageno 798
nowidow 799
ntheorem 149, 799
overpic 153, 812
pagegrid 813
pagenote 129, 813
pagesel 813
paralist 813
parnotes 814
parskip 815
pbox 815
pdflscape 816
pdfpages 816
pdfrender 819
pdfsync 819
pdftricks 152, 820
pdfx 820
pfnote 821
phfqit 821
placeins 822
polyglossia 161
prelim2e 822
prettyref 822
preview 822
printlen 207
pst-eps 823
pstricks 152, 823
quotchap 824
quoting 825
ragged2e 825
realscripts 826
refcount 206
register 827
relsize 120, 828
repeatindex 829
resizegather 830
romanbar 830
romanbarpagenumber 831
rotating 831
rotfloat 831
savetrees 832
scalefnt 833
schemata 833

scrextend 834
scrhack 837
scrlayer 837
scrlayer-notecolumn 839
scrlayer-scrpage 839
section 840
sectionbreak 841
sectsty 842
setspace 842
shadow 844
showidx 844
showkeys 844
sidecap 845
sidenotes 846
SIunits 847
siunitx 149, 530, 848
soul 853
soulpos 855
soulutf8 856
splitidx 856
stabular 858
stfloats 859
subcaption 158
subfig 158, 859
subfigure 864
supertabular 157, 865
syntonly 866
t1enc 867
tabls 867
tabularx 867
tabulary 868
textarea 868
textcomp 101, 119, 869
textfit 872
textpos 872
theorem 873
threeparttable 877
thumb 878
thumbs 878
tikz 151, 879
titleps 880
titleref 884
titlesec 884
titletoc 886
titling 128, 888
tocbasic 893
tocbibind 144, 893, 894
tocenter 895
tocloft 128, 144, 598, 896
tocstyle 902
todo 903
todonotes 162, 904
transparent 906
trimclip 906
trivfloat 158, 907
turnthepage 908
twoup-gen 908
typearea 909
ulem 909
underscore 912
units 149, 912
upref 912
url 125, 913
varioref 125
varwidth 207
verbatim 206
verse 160, 913, 914
vertbars 915
vmargin 915
vowel 916
vwcol 917
wallpaper 919
wasysym 920
watermark 920
widows-and-orphans 920
wrapfig 921
xcolor 152, 533, 922
xellipsis 933
xfrac 934
xifthen 206
xltabular 937
xltxtra 937
xmpincl 938
xparse 162, 205
xpatch 187
xpiano 938
xstring 206
xtab 157, 939
xunicode 941
xurl 941
xy 941
zref 207
zwpagelayout 942
\packagediagramname 111, 8614
page
counter 91, 179
inacessible 108
\pagebreak 10220
\pagecolor 71
pagegrid (package) 813

pagenote (package) *129*, *813*
\pagenumbering 4569
\pageref 7983
\pagerefPageFor 7982
pagesel (package) *813*
\pagestyle 4561
Pandoc (program) *75*
\par
  hooks 325
\paragraph 5658
paralist (package) *813*
\parbox 9789
parnotes (package) *814*
\parsemulticolumnalignment 7333
parskip (package) *815*
\part 5589
pbox (package) *815*
PDF images
  converting 97
  using 150, 531
pdfcrop (program) [requirement] *78*
pdfLaTeX (program) [requirement] . . . *78*
pdflscape (package) *816*
pdfpages (package) *816*
pdfrender (package) *819*
pdfseparate (program) [requirement]
  . . . . . *78*, *83*
pdfsync (package) *819*
pdftocairo (program) *150*, *531*
pdftocairo (program) [requirement] *78*, *83*
pdftotext (program) [requirement] *78*, *83*
pdftotextEnc (option) *105*, *190*
pdftricks (package) *152*, *820*
pdfx (package) *820*
Perl 84
perl (program) [requirement] *84*
pfnote (package) *821*
\phantomsection 10281
phfqit (package) *821*
picture (environment) *541*, 9681
placeins (package) *822*
Plastex (program) *74*
\PN@parnotes@auto 4888
PNG images 150, 532
polyglossia (package) *161*
\popclose 4352
Poppler 78, 83
prelim2e (package) *822*
prettyref (package) *822*
preview (package) *822*
print
  selecting print/HTML definitions . . 215
\printauthor *364*, 5882, 5901
\printdate *364*, 5893, 5903
printglossary (option) [lwarpmk] *129*, *713*
\printindex 2
PrintIndexCmd (option) *102*, *191*
printlen (package) *207*
\printthanks *363*, *366*
\printtitle *364*, 5866, 5900
problems 176
program:
  [requirement]:
    LuaLaTeX *78*
    MathJax *78*
    pdfcrop *78*
    pdfLaTeX *78*
    pdfseparate *78*, *83*
    pdftocairo *78*, *83*
    pdftotext *78*, *83*
    perl *84*
    XeLaTeX *78*
  Adobe *75*
  AsciiDoc *75*
  AsciiDoctor *75*
  Asciidoctor-LaTeX *75*
  epstopdf *150*, *532*
  Flare *75*
  FrameMaker *75*
  GELLMU *74*
  GladTeX *74*
  Hevea *74*
  htlatex *74*
  InDesign *75*
  LaTeX2HTML *74*
  LaTeXML *74*
  LibreOffice *75*
  Linux *113*, *187*
  lwarpmk *175*, *279*
  lwarpmk epstopdf *150*, *532*
  lwarpmk pdftosvg *150*, *532*
  Mac OS *113*, *187*
  Madcap *75*
  makeindex *141*
  MathJax *146*, *490*, *491*
  MS-Windows *114*, *187*
  OpenOffice *75*
  Pandoc *75*
  pdftocairo *150*, *531*
  Plastex *74*

TeX2page 74
TeX4ht 74
TeXMaths 168
TtH 74
Unix 113, 187
Windows 114, 187
Word 75
xindy 142
programs
utility 77
project.css (file) 113
project.lwarpmkconf (file) 234
project_html.tex (file) 227
pst-eps (package) 823
pstricks (package) 152, 823
published 372
\pushclose 4325

Q

\qquad 561, 10169
\quad 561, 10164
quotchap (package) 824
quote (environment) 6042
quoting (package) 825

R

ragged2e (package) 825
\raggedbottom 4565
\raggedleft 9423
\raggedright 9429
\raisebox 9918
realscripts (package) 826
\ref 7971
refcount (package) 206
\reflectbox 312
register (package) 827
relsize (package) 120, 828
repeatindex (package) 829
[requirement]:
LuaLaTeX (program) 78
MathJax (program) 78
pdfcrop (program) 78
pdfLaTeX (program) 78
pdfseparate (program) 78, 83
pdftocairo (program) 78, 83
pdftotext (program) 78, 83
perl (program) 84
XeLaTeX (program) 78
\RequirePackage 706
\resizebox 321
resizegather (package) 830
\ResumeTabular 7594
\reversemarginpar 5180
\rightline 9437
\rmfamily 10059
romanbar (package) 830
romanbarpagenumber (package) 831
\rotatebox 278
rotating (package) 831
rotfloat (package) 831
\rowcolor 6801
\rownum 6793
\rule 10227

S

sample_project.css (file) 113, 273
savetrees (package) 832
\sb 10067
\scalebox 297
scalefnt (package) 833
schemata (package) 833
scrextend (package) 834
scrhack (package) 837
scrlayer (package) 837
scrlayer-notecolumn (package) 839
scrlayer-scrpage (package) 839
\scshape 10064
section
depths 182
heading, word processor 168
missing 108
\section 5628
section (package) 840
sectionbreak (package) 841
sectsty (package) 842
\SetHTMLFileNumber 4453
setspace (package) 842
settings
css project-specific 113
CSS selection 112
HTML conversion 106
lwarp package options 102
selecting output 114
title page 115
\sffamily 10060
\sfrac 934
shadow (package) 844
shell escape 100
showidx (package) 844
showkeys (package) 844

sidecap (package) *845*
sidenotes (package) *846*
SideTOCDepth (counter) *108*, *479*
\sidetocname 8359
\simplechapterdelim 5438
SIunits (package) *847*
siunitx
  with **TeXMaths** 168
siunitx (package) *149*, *530*, *848*
\sloppy 4567
soul (package) *853*
soulpos (package) *855*
soulutf8 (package) *856*
\sp 10066
space
  horizontal 118
    between minipages 561
split
  miss-numbered 149, 799
splitidx (package) *856*
stabular (package) *858*
stack depths 182
\StartDefiningMath 4400
\StartDefiningTabulars 4390
\StartDynamicMath 4412
stfloats (package) *859*
\StopDefiningMath 4404
\StopDefiningTabulars 4394
\StopDynamicMath 4416
subcaption (package) *158*
subequations
  MathJax 147, 491
subfig (package) *158*, *859*
subfigure (package) *864*
\subparagraph 5666
\subsection 5641
\subsubsection 5649
subtitle 372
supertabular (package) *157*, *865*
SVG
  mathsvg option 189
  converting from PDF or EPS 97
  dynamic math 302
  image processing 279
  images appearing as HTMLcode 91, 179
  math incorrect images 302
  math summary 145, 490
syntonly (package) *866*

T

t1enc (package) *867*
tabbing
  and math 153
tabbing (environment) 6126
table
  text alignment 157
\tableofcontents *110*, 8301
tabls (package) *867*
tabular
  baseline 408
  HTML columnn conversion 408
  macros inside 155, 390
  misplaced \noalign 155, 390
  misplaced alignment tab character & 153, 388
  multicolumn with multirow 792
  row corruption 155, 390
\TabularMacro 7592
tabularx (package) *867*
tabulary (package) *868*
tagged equations
  Mathjax 146, 491
\TeX 10289
TeX2page (program) *74*
TeX4ht (program) *74*
TeXMaths (program) *168*
textarea (package) *868*
\textbf 9941
\textcolor 48
textcomp
  missing symbols 119
textcomp (package) *101*, *119*, *869*
textfit (package) *872*
\textgreater 4443
\textit 9966
\textless 4443
\textmd 9935
\textnormal 9983
textpos (package) *872*
\textrm 9944
\textsc 9969
\textsf 9951
\textsl 9976
\textsubscript 10070
\textsuperscript 10068
\texttt 9956
\textup 9959
\tfl@chapter@fix *907*
\thanks *116*

\thanksmarkseries 91
\theauthor *364*
thebibliography (environment) . . . 8525
\thedate *364*
theindex (environment) 8438
\thempfootnote 5122
theorem (package) *873*
\thetitle *364*
\thispagestyle 4562
threeparttable (package) *877*
thumb (package) *878*
thumbs (package) *878*
tikz
  catcodes 879
  dollar redefined 879
  in display math 148
tikz (package) *151*, *879*
Title
  HTML meta tag 328
title
  HTML meta tag 117, 328
\title *116*, 5023
titlepage
  subtitle and published 372
titlepage (environment) *116*, 5850
titleps (package) *880*
titleref (package) *884*
titles (option) [tocloft] *128*
titlesec (package) *884*
titletoc (package) *886*
titling (package) *128*, *888*
titlingpage (environment) 14, *116*
tocbasic (package) *893*
[tocbibind]:
  numindex (option) *144*, *894*
tocbibind (package) *144*, *893*, *894*
tocdepth (counter) *108*
tocenter (package) *895*
[tocloft]:
  titles (option) *128*
tocloft (package) *128*, *144*, *598*, *896*
tocstyle (package) *902*
todo (package) *903*
todonotes (package) *162*, *904*
\tracinglwarp *180*, 826
transparent (package) *906*
trimclip (package) *906*
trivfloat
  with newfloat, algorithmic 908
trivfloat (package) *158*, *907*
troubleshooting 176
  HTML debug comments 214
\ttfamily 10061
TtH (program) *74*
turnthepage (package) *908*
tutorial.tex (file) *85*
twoup-gen (package) *908*
typearea (package) *909*

U

ulem (package) *909*
\underline 10078
underscore
  filename 106, 120
underscore (package) *912*
Unicode
  enhanced coverage 100
  file & section names 344
  input characters 185
  selection 100
units (package) *149*, *912*
Unix (program) *113*, *187*
\up 10076
upref (package) *912*
\upshape 10062
\url 8062
url (package) *125*, *913*
usingOSWindows (boolean) *187*
UTF-8
  enhanced coverage 100
  file & section names 344
  index 101
  selection 100
utility
  programs 77

V

varioref (package) *125*
varwidth (package) *207*
verbatim
  footnotes 126
verbatim (environment) 6116
verbatim (package) *206*
\VerbatimHTMLWidth (length) *376*
\verbatiminput 6108
verse (environment) 2, 6057
verse (package) *160*, *913*, *914*
vertbars (package) *915*
viewport
  HTML meta tag 357

\vleftmargini (length) 160, 375, 913
\vleftskip (length) . . . 160, 375, 913, 914
vmargin (package) 915
vowel (package) 916
vwcol (package) 917

W

wallpaper (package) 919
warning icon 183
warpall (environment) 114, 208
warpHTML (environment) 101, 111, 114, 209
warpHTML (option) 105, 189
\warpHTMLonly 111, 115, 198
warpingHTML (boolean) 188
warpingprint (boolean) 188
warpprint (environment) . . 111, 114, 209
warpprint (option) 105, 189
\warpprintonly 111, 115, 197
wasysym (package) 920
watermark (package) 920
widows-and-orphans (package) 920
Windows (program) 114, 187
Word (program) 75
word processor
    conversion recommendations 168
    HTML conversion settings . . . 165, 220
    section headings 168
WPMarkFloats (boolean) 167, 220
WPMarkLOFT (boolean) 168, 221
WPMarkMath (boolean) 168, 221
WPMarkMinipages (boolean) 167, 220
WPMarkTOC (boolean) 167, 220
WPTitleHeading (boolean) 168, 221
wrapfig (package) 921

X

xcolor (package) 152, 533, 922
XeLaTeX
    detection 184
    file & section names 344
\XeLaTeX 10321
XeLaTeX (program) [requirement] 78
xellipsis (package) 933
\XeTeX 10321
xfrac (package) 934
\xfracHTMLfontsize 3
xifthen (package) 206
xindy
    and hyperref 131
    customizing 142
xindy (option) 102, 106, 191
xindy (program) 142
xindyCodepage (option) 102, 190
xindyLanguage (option) 102, 190
xindyStyle (option) 102, 142, 190
xltabular (package) 937
xltxtra (package) 937
xmpincl (package) 938
xparse
    warnings 162
xparse (package) 162, 205
xpatch (package) 187
xpiano (package) 938
xstring (package) 206
xtab (package) 157, 939
xunicode (package) 941
xurl (package) 941
xy (package) 941

Z

zref (package) 207
zwpagelayout (package) 942

For the most recent changes, followed by the start of the Index, see page 997.