



The l warp package

LATEX to HTML

v0.39 — 2017/09/05

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

The l warp package allows LATEX to directly produce HTML 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 l warp version of a package for HTML when available. More than a hundred 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 1, Updates.

Note that this is still an “alpha” version of l warp, 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.

Contents

1	l warp.sty	27
1	Updates	27
2	Introduction	33
2.1	Supported packages and features	34
3	Alternatives	38
3.1	Internet class	38
3.2	TeX4ht	38
3.3	Translators	38
3.4	AsciiDoc	39
3.5	Pandoc	39
3.6	Word processors	39
3.7	Commercial systems	40
3.8	Comparisons	40
4	Installation	41
4.1	Installing the l warp package	41
4.2	Installing the l warpmk utility	44
4.2.1	Using a local copy of l warpmk	45
4.3	Installing additional utilities	46
5	Tutorial	48
5.1	Starting a new project	48
5.2	Compiling the print version with l warpmk	52
5.3	Compiling the HTML version with l warpmk	53
5.4	Generating the SVG images	54
5.5	Using MathJax for math	55
5.6	Changing the CSS style	56
5.7	Customizing the HTML output	57
5.8	Using latexmk	62
5.9	Using XeLaTeX or LuaLaTeX	63
5.10	Using a glossary	64
5.11	Cleaning auxiliary files	65
5.12	Cleaning auxiliary and output files	65
5.13	Processing multiple projects in the same directory	65
5.14	Using the make utility	65
6	Additional details	66
6.1	Font and UTF-8 support	66
6.1.1	Indexes and UTF-8	67
6.2	l warp package loading and options	67
6.3	Selecting the operating system	68

6.4	Selecting actions for print or HTML output	69
6.5	Commands to be placed into the <code>warpprint</code> environment	70
6.6	Title page	70
6.7	HTML page meta descriptions	71
6.8	HTML page meta author	71
6.9	CSS	72
6.10	Modifying <code>xindy</code> index processing	72
7	Special cases and limitations	73
7.1	Formatting	73
7.1.1	Text formatting	73
7.1.2	Horizontal space	73
7.1.3	Text alignment	73
7.1.4	Accents	74
7.1.5	<code>Textcomp</code>	74
7.1.6	Superscripts and other non-math uses of math mode	74
7.1.7	Empty <code>\item</code> followed by a new line of text or a nested list:	74
7.1.8	Filenames in lists	74
7.1.9	<code>relsize</code> package	74
7.2	Boxes and minipages	75
7.2.1	Save Boxes	75
7.2.2	Minipages	75
7.2.3	Side-by-side minipages	76
7.2.4	Framed minipages and other environments	76
7.2.5	<code>fancybox</code> package	77
7.2.6	<code>mdframed</code> package	78
7.3	Cross-references	79
7.3.1	Page references	79
7.3.2	<code>cleveref</code> and <code>variorref</code> packages	79
7.3.3	Hyperlinks, <code>hyperref</code> , and <code>url</code>	79
7.3.4	Footnotes and page notes	80
7.4	Front and back matter	80
7.4.1	Starred chapters and sections	80
7.4.2	<code>abstract</code> package	81
7.4.3	<code>titling</code> and <code>authblk</code>	81
7.4.4	<code>appendix</code> package	81
7.4.5	<code>pagenote</code> package	81
7.4.6	<code>endnotes</code> package	81
7.4.7	<code>glossaries</code> package	82
7.4.8	Index and the <code>tocbibind</code> package	82
7.5	Math	83
7.5.1	<code>ntheorem</code> package	85
7.5.2	<code>siunitx</code> package	85
7.5.3	<code>units</code> and <code>nicefrac</code> packages	86
7.5.4	<code>newtxmath</code> package	86
7.6	Graphics	86

7.6.1	grffile package	87
7.6.2	color package	87
7.6.3	xcolor package	87
7.7	Tabular	88
7.7.1	longtable package	90
7.7.2	supertabular and xtab packages	90
7.7.3	float, trivfloat, and/or algorithmicx together	91
7.7.4	caption and subcaption packages	91
7.8	Floats	91
7.8.1	subfig package	91
7.8.2	floatrow package	91
7.9	Miscellaneous	92
7.9.1	verse and memoir	92
7.9.2	newclude package	92
7.9.3	babel package	93
8	EPUB conversion	94
9	Word-processor conversion	96
10	Modifying l warp	98
10.1	Creating an l warp version of a package	98
10.2	Adding a package to the l warp.dtx file	99
10.3	Testing l warp	99
10.4	Modifying l warpmk	99
11	Troubleshooting	101
11.1	Using the l warp.sty package	101
11.1.1	Debug tracing output	103
11.2	Compiling the l warp.dtx file	103
12	Implementation	104
13	Stack depths	104
14	Source Code	106
15	Detecting the TeX Engine — pdflatex, lualatex, xelatex	107
16	Unicode Input Characters	107
17	Early package requirements	108
18	Operating-System portability	109
18.1	Common portability code	109
18.2	Unix, Linux, and Mac OS	109
18.3	MS-Windows	109

19 Package options	110
19.1 Conditional compilation	113
20 Misplaced packages	114
21 Required packages	118
22 Loading packages	123
23 File handles	125
24 Include a file	126
25 Copying a file	127
26 Debugging messages	128
27 HTML-conversion output modifications	129
28 Remembering original formatting macros	129
29 Accents	131
30 Configuration Files	133
30.1 project_html.tex	133
30.2 lwarpmk.conf	133
30.3 project.lwarpmkconf	134
30.4 l warp.css	135
30.5 l warp_sagebrush.css	161
30.6 l warp_formal.css	166
30.7 sample_project.css	170
30.8 l warp.xdy	171
30.9 l warp_mathjax.txt	171
30.10 lwarpmk option	173
31 Stacks	184
31.1 Assigning depths	185
31.2 Closing actions	185
31.3 Closing depths	186
31.4 Pushing and popping the stack	186
32 Data arrays	188
33 HTML entities	188
34 HTML filename generation	189
35 Homepage link	192

36	\PrintStack diagnostic tool	193
37	Closing stack levels	193
38	PDF pages and styles	194
39	HTML tags, spans, divs, elements	195
39.1	Mapping L ^A T _E X Sections to HTML Sections	195
39.2	Babel-French	196
39.3	HTML tags	197
39.4	Block tags and comments	199
39.5	Div class and element class	200
39.6	Single-line elements	201
39.7	HTML5 semantic elements	202
39.8	High-level block and inline classes	202
39.9	Closing HTML tags	203
40	Paragraph handling	204
41	Paragraph start/stop handling	208
42	Page headers and footers	210
43	CSS	211
44	HTML meta description and author	212
45	Footnotes	213
45.1	Regular page footnotes	213
45.2	Minipage footnotes	213
45.3	Titlepage thanks	213
45.4	Regular page footnote implementation	213
45.5	Minipage footnote implementation	215
45.6	Printing pending footnotes	216
46	Marginpars	217
47	Splitting HTML files	217
48	Sectioning	222
48.1	User-level starred section commands	223
48.2	Book class commands	223
48.3	Sectioning support macros	224
48.4	\section and friends	230
49	Starting a new file	232
50	Starting HTML output	235

51 Ending HTML output	238
52 Title page	239
52.1 Setting the title, etc.	240
52.2 Changes for \affiliation	241
52.3 Printing the thanks	242
52.4 Printing the title, etc. in HTML	242
52.5 Printing the title, etc. in print form	244
52.6 \maketitle for HTML output	244
52.7 \published and \subtitle	247
53 Abstract	248
54 Quote and verse	249
54.1 Citations and attributions	249
54.2 Quotes, quotations	250
54.3 Verse	250
55 Verbatim	251
56 Theorems	253
57 Lists	254
57.1 Itemize	255
57.2 Enumerate	256
57.3 Description	256
58 Tabular	258
58.1 Limitations	258
58.2 Token lookahead	260
58.3 Booleans	261
58.4 Handling &, @, and !	262
58.4.1 Localizing & catcodes	263
58.4.2 Handling &	264
58.5 Handling \\	265
58.6 Variables	266
58.7 Parsing @, >, <, ! columns	267
58.8 Parsing 'l', 'c', or 'r' columns	269
58.9 Parsing 'p', 'm', or 'b' columns	270
58.10 Parsing 'D' columns	270
58.11 Parsing the column specifications	271
58.12 Starting a new row	275
58.13 Printing at or bang tags	275
58.14 Data opening tag	276
58.15 Midrules	277
58.16 Multicolumns	278
58.16.1 Parsing multicolumns	278

58.16.2 Multicolumn factored code	281
58.16.3 Multicolumn	282
58.16.4 Longtable captions	283
58.16.5 Counting HTML tabular columns	285
58.17 Multicolumnrow	287
58.18 Utility macros inside a table	287
58.19 Special-case tabular markers	288
58.20 Checking for a new table cell	289
58.21 \mrowcell	290
58.22 New \tabular definition	291
58.23 Array	294
59 Cross-references	294
59.1 Setup	294
59.2 Zref setup	296
59.3 Labels	297
59.4 References	299
59.5 Hyper-references	301
60 Floats	304
60.1 Float captions	305
60.1.1 Caption inside a float environment	306
60.1.2 Caption and LOF linking and tracking	307
61 Table of Contents, LOF, LOT	310
61.1 Reading and printing the TOC	310
61.2 High-level TOC commands	312
61.3 Side TOC	313
61.4 Low-level TOC line formatting	314
62 Index and glossary	317
63 Math	319
63.1 Inline and display math	321
63.2 MathJax support	323
63.3 Equation environment	327
63.4 AMS Math environments	329
63.4.1 Support macros	329
63.4.2 Environment patches	330
64 Lateximages	337
65 center, flushleft, flushright	343
66 Textcomp	344
67 Siunitx	344

68 Graphics	345
68.1 \graphicspath	346
68.2 Length conversions and graphics options	346
68.3 \includegraphics	349
68.4 \rotatebox, \scalebox, \reflectbox	353
68.5 Null functions	356
68.6 xcolor print-mode additions	356
69 Cleveref	360
70 Picture	361
71 Boxes and Minipages	362
71.1 Counters and lengths	362
71.2 Footnote handling	363
71.3 Minipage handling	363
71.4 Parbox, makebox, framebox, fbox, raisebox	366
72 Direct formatting	372
73 Skips, spaces, font sizes	376
74 \phantomsection	383
75 \LaTeX and other logos	384
75.1 HTML logos	384
75.2 Print logos	386
76 \AtBeginDocument, \AtEndDocument	386
77 Trademarks	387
2 l warp-a4.sty	388
78 a4	388
3 l warp-a4wide.sty	388
79 a4wide	388
4 l warp-a5comb.sty	388
80 a5comb	388

5 l warp-abstract.sty	388
81 abstract	388
6 l warp-addlines.sty	390
82 addlines	390
7 l warp-afterpage.sty	390
83 afterpage	390
8 l warp-algorithmicx.sty	391
84 algorithmicx	391
9 l warp-alltt.sty	391
85 alltt	391
10 l warp-amsthm.sty	392
86 amsthm	392
11 l warp-any size.sty	394
87 anysize	394
12 l warp-appendix.sty	395
88 appendix	395
13 l warp-arabicfront.sty	395
89 arabicfront	395
14 l warp-authblk.sty	395
90 authblk	395

15 l warp-balance.sty	396
91 balance	396
16 l warp-bigdelim.sty	397
92 bigdelim	397
17 l warp-bigstrut.sty	397
93 bigstrut	397
18 l warp-bookmark.sty	398
94 bookmark	398
19 l warp-booktabs.sty	398
95 booktabs	398
20 l warp-boxedminipage.sty	399
96 boxedminipage	399
21 l warp-boxedminipage2e.sty	399
97 boxedminipage2e	399
22 l warp-caption2.sty	399
98 caption2	399
23 l warp-ccaption.sty	400
99 ccaption	400
24 l warp-changepage.sty	400
100 changepage	400

25 lwarf-chngpage.sty	400
101 chngpage	400
26 lwarf-chappg.sty	400
102 chappg	400
27 lwarf-color.sty	401
103 color	401
28 lwarf-crop.sty	401
104 crop	401
29 lwarf-cutwin.sty	401
105 cutwin	401
30 lwarf-dcolumn.sty	402
106 dcolumn	402
31 lwarf-draftwatermark.sty	402
107 draftwatermark	402
32 lwarf-ebook.sty	403
108 ebook	403
33 lwarf-ellipsis.sty	403
109 ellipsis	403
34 lwarf-emptypage.sty	403
110 emptypage	403

35 lwarf-endnotes.sty	404
111 endnotes	404
36 lwarf-enumerate.sty	404
112 enumerate	404
37 lwarf-enumitem.sty	405
113 enumitem	405
38 lwarf-epigraph.sty	405
114 epigraph	405
39 lwarf-eso-pic.sty	406
115 eso-pic	406
40 lwarf-everypage.sty	407
116 everypage	407
41 lwarf-everyshi.sty	407
117 everyshi	407
42 lwarf-extramarks.sty	408
118 extramarks	408
43 lwarf-fancybox.sty	408
119 fancybox	408
44 lwarf-fancyhdr.sty	412
120 fancyhdr	412

45 l warp-fancyvrb.sty	413
121 fancyvrb	413
46 l warp-figcaps.sty	418
122 figcaps	418
47 l warp-fix2col.sty	418
123 fix2col	418
48 l warp-float.sty	419
124 float and \newfloat	419
49 l warp-floatflt.sty	420
125 floatflt	420
50 l warp-floatpag.sty	421
126 floatpag	421
51 l warp-floatrow.sty	421
127 floatrow	421
52 l warp-flushend.sty	426
128 flushend	426
53 l warp-fncychap.sty	427
129 fncychap	427
54 l warp-fontenc.sty	428
130 fontenc	428

55 l warp-fontspec.sty	428
131 fontspec	428
56 l warp-footmisc.sty	428
132 footmisc	428
57 l warp-footnote.sty	429
133 footnote	429
58 l warp-footnotehyper.sty	430
134 footnotehyper	430
59 l warp-framed.sty	431
135 framed	431
60 l warp-ftnright.sty	433
136 ftnright	433
61 l warp-fullpage.sty	434
137 fullpage	434
62 l warp-fullwidth.sty	434
138 fullwidth	434
63 l warp-geometry.sty	434
139 geometry	434
64 l warp-glossaries.sty	435
140 glossaries	435

65 l warp-graphics.sty	436
141 graphics	436
66 l warp-graphicx.sty	436
142 graphicx	436
67 l warp-grffile.sty	437
143 grffile	437
68 l warp-hyperref.sty	437
144 hyperref	437
69 l warp-hyperxmp.sty	440
145 hyperxmp	440
70 l warp-idxlayout.sty	440
146 idxlayout	440
71 l warp-indentfirst.sty	441
147 indentfirst	441
72 l warp-inputenc.sty	441
148 inputenc	441
73 l warp-keyfloat.sty	442
149 keyfloat	442

74 lwarf-layout.sty	443
150 layout	443
75 lwarf-letterspace.sty	443
151 letterspace	443
76 lwarf-lettrine.sty	444
152 lettrine	444
77 lwarf-lips.sty	445
153 lips	445
78 lwarf-listings.sty	445
154 listings	445
79 lwarf-longtable.sty	449
155 longtable	449
80 lwarf-lscape.sty	451
156 lscape	451
81 lwarf-ltcaption.sty	451
157 ltcaption	451
82 lwarf-marginfit.sty	451
158 marginfit	451

83 lwarf-marginfix.sty	452
159 marginfix	452
84 lwarf-marginnote.sty	452
160 marginnote	452
85 lwarf-mcaption.sty	453
161 mcaption	453
86 lwarf-mdframed.sty	453
162 mdframed	453
87 lwarf-metaprolog.sty	462
163 metaprolog	462
88 lwarf-microtype.sty	462
164 microtype	462
89 lwarf-moreverb.sty	463
165 moreverb	463
90 lwarf-mparhack.sty	465
166 mparhack	465
91 lwarf-multicol.sty	465
167 multicol	465
92 lwarf-multirow.sty	466
168 multirow	466

168.1 Multirow	467
168.2 Combined multicolumn and multirow	468
93 lwarf-nameref.sty	469
169 nameref	469
94 lwarf-needspace.sty	469
170 needspace	469
95 lwarf-newinclude.sty	469
171 newinclude	469
96 lwarf-newunicodechar.sty	470
172 newunicodechar	470
97 lwarf-nextpage.sty	470
173 nextpage	470
98 lwarf-nonumonpart.sty	470
174 nonumonpart	470
99 lwarf-nopageno.sty	471
175 nopageno	471
100 lwarf-nowidow.sty	471
176 nowidow	471
101 lwarf-ntheorem.sty	471
177 ntheorem	471

102 l warp-pagenote.sty	482
178 pagenote	482
 103 l warp-paralist.sty	482
179 paralist	482
 104 l warp-parskip.sty	483
180parskip	483
 105 l warp-pdflandscape.sty	483
181 pdflandscape	483
 106 l warp-pdfsync.sty	483
182 pdfsync	483
 107 l warp-placeins.sty	484
183 placeins	484
 108 l warp-prelim2e.sty	484
184 prelim2e	484
 109 l warp-ragged2e.sty	484
185 ragged2e	484
 110 l warp-realscripts.sty	485
186 realscripts	485

111 l warp-relsize.sty	486
187 relsize	486
112 l warp-romanbar.sty	487
188 romanbar	487
113 l warp-romanbarpagenumber.sty	488
189 romanbarpagenumber	488
114 l warp-rotating.sty	488
190 rotating	488
115 l warp-rotfloat.sty	488
191 rotfloat	488
116 l warp-savetrees.sty	489
192 savetrees	489
117 l warp-scalefnt.sty	489
193 scalefnt	489
118 l warp-setspace.sty	490
194 setspace	490
119 l warp-shadow.sty	491
195 shadow	491
120 l warp-showidx.sty	491
196 showidx	491

121 l warp-showkeys.sty	491
197 showkeys	491
122 l warp-sidecap.sty	492
198 sidecap	492
123 l warp-sidenotes.sty	492
199 sidenotes	492
124 l warp-siunitx.sty	494
200 siunitx	494
125 l warp-soul.sty	494
201 soul	494
126 l warp-subfig.sty	496
202 subfig	496
127 l warp-supertabular.sty	502
203 supertabular	502
128 l warp-syntonly.sty	503
204 syntonly	503
129 l warp-tabularx.sty	504
205 tabularx	504
130 l warp-tabulary.sty	504
206 tabulary	504

131 lwarf-textarea.sty	505
207 textarea	505
132 lwarf-textcomp.sty	505
208 textcomp	505
133 lwarf-textpos.sty	507
209 textpos	507
134 lwarf-theorem.sty	507
210 theorem	507
135 lwarf-threeparttable.sty	511
211 threeparttable	511
136 lwarf-tikz.sty	512
212 tikz	512
137 lwarf-titleps.sty	513
213 titleps	513
138 lwarf-titleref.sty	515
214 titleref	515
139 lwarf-titlesec.sty	515
215 titlesec	515
140 lwarf-titletoc.sty	516
216 titletoc	516

141 l warp-titling.sty	517
217 titling	517
142 l warp-toctbibind.sty	521
218 toctbibind	521
143 l warp-tocloft.sty	522
219 tocloft	522
144 l warp-transparent.sty	527
220 transparent	527
145 l warp-trivfloat.sty	527
221 trivfloat	527
221.1 Combining \newfloat, \trivfloat, and algorithmicx	528
146 l warp-typearea.sty	528
222 typearea	528
147 l warp-ulem.sty	529
223 ulem	529
148 l warp-upref.sty	530
224 upref	530
149 l warp-verse.sty	531
225 verse	531

150 l warp-wallpaper.sty	532
226 wallpaper	532
151 l warp-wrapfig.sty	533
227 wrapfig	533
152 l warp-xcolor.sty	534
228 xcolor	534
228.1 Xcolor definitions: location and timing	535
228.2 Code	537
153 l warp-xfrac.sty	542
229 xfrac	542
154 l warp-xltextra.sty	545
230 xltextra	545
155 l warp-xmpincl.sty	545
231 xmpincl	545
156 l warp-xtab.sty	546
232 xtab	546
157 l warp-zwpagelayout.sty	547
233 zwpagelayout	547
Change History and Index	550

List of Figures

1	tutorial.tex listing	49
---	----------------------	----

List of Tables

1	l ^A T _E X–HTML generation — l warp package — Supported functions	34
2	Required software programs	42
3	Files created along with the print version	51
4	Package options	68
5	Section depths and HTML headings	105
6	Tabular baseline	271
7	Tabular HTML column conversions	272
8	Cross-referencing data structures	295
9	Float data structures	304

File 1 **lwarf.sty**

1 Updates

The following is intended for those updating existing projects which use lwarf, highlighting any special changes which must be made due to improvements or modifications in lwarf itself.

For a detailed list of changes, see the Change History on page 550.

v0.39:

title pages

⚠ \published and
\subtitle

⚠ load order
tabular

multi column/row cell

⚠ macros inside tabular

⚠ tabular defined inside
another environment

- 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 52.7. authblk is added, and should be loaded before titling. See section 52.
- \multirow now supports the new optional vpos argument.
- Added \multicolumnrow for combined \multicolumn and \multirow. See section 168.2.
- Tabular special cases:
 - Added \TabularMacro to mark custom macros inside tabular data cells, avoiding row corruption. See section 7.7.
 - Added \ResumeTabular for use when a tabular environment is defined inside another environment. See section 7.7.
- Added supertabular, xtab, bigstrut, bigdelim.
- Added fullwidth.
- Added addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, typearea, ebook.

v0.38:

forced single-pass compile

starred sections

- 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.
- Added \ForceHTMLPage and \ForceHTMLTOC to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 7.4.1.

- [updated tutorial](#)
 - Modified the tutorial to use the new \ForceHTMLPage and \ForceHTMLTOC macros.

- [packages](#)
 - Added appendix, tocbibind, fncychap, fix2col.

- [packages](#)
 - Added relsize, scalefnt.
 - Added realscripts, metalogo, xltextra.
 - Added grffile, romanbar.
 - Added arabicfront, chappg, nonumonpart, nopageno, romanbarpagenum-

- [front & back matter](#)
 - Docs: Improved description of the use of front/back matter. See section [7.4](#).

- [front & back matter](#)
 - 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 lwarf, now maintains independent cut files for print and HTML versions, helping latexmk to better know whether to recompile.
- [accents and symbols](#)
 - Improved support for L^ET_X accents, textcomp, siunitx symbols.
- [babel-french](#)
 - Improved babel-french handling for load order and ~ tilde.

v0.36:

- [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**
- 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 7.2.2. `\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 7.2.4. `\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.
 - Added the `eqnarray` environments.
 - `fancyvrb` is no longer required (preloaded), but is still supported.
 - Added `verbatim` and `moreverb`.
 - Added `fancybox`, `boxedminipage2e` and `shadow`.
 - `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.
 - Adds fixed-width HTML spaces around punctuation when using `babel-french`. \LaTeX does not yet use the extra punctuation spacing.
- eqnarray**
- verbatim packages**
- framing packages**
- list packages**
- babel-french**

v0.33:

- Tabular @ and ! columns now have their 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 \EndDefiningTabulars. See section 58.4.1. 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 (new)	IndexLanguage xdyFilename
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 lwarf, such as:

```
\usepackage[
    HomeHTMLfilename=index,
    HTMLfilename={},
    IndexLanguage=english
]{lwarf}
```

- The file lwarf_html.xdy has been renamed lwarf.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 lwarf.xdy.
 3. The old file lwarf_html.xdy may be deleted.
- The new lwarf package option xdyFilename may be used to tell lwarpmk to use a custom .xdy file instead of lwarf.xdy. See section 6.10.
- 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 `lwarf-newproject` and `lwarf`. Sets the language to use while processing the glossary. (As of v0.30, this has been changed to the `IndexLanguage` option.)
- Fix: `\includegraphics` when no optional arguments.

v0.28:

- `\HTMLAuthor {\<name>}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Boolean `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.
- Boolean `HTMLMarkFloats` adds text marks around floats only if `FormatWordProcessor`. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames.
- 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 `printglossary` and `htmlglossary` actions.

2 Introduction

The lwarp project aims to allow a rich \LaTeX document to be converted to a reasonable HTML interpretation. 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 HTML, CSS, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers show that these new features degrade gracefully, although the SVG format for math may not be available on small cell phones.

`pdflatex`, `xelatex`, or `lualatex` is 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 JPG are used as-is.

SVG images may be used for math, and are also used for `picture`, and `Tikz` environments, as this format has better browser and e-book support than MathML (as of this writing), while still allowing for the high-quality display and printing of images (again, subject to potentially bug-ridden¹ 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.² Custom \LaTeX macros may be used as-is in math expressions, since the math is evaluated entirely inside \LaTeX .

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, one per math expression. lwarp maintains \LaTeX control for cross-referencing and equation numbering / formatting.

The lwarp package allows \LaTeX to directly generate HTML tags from a \LaTeX source document, with only minor intervention on the user’s part. 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 HTML tags, to a number of HTML plain-text files and accompanying images.

¹Firefox has had an on-again/off-again bug for quite some time regarding printing svgs at high resolution.

²There 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!

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 usual \TeX version. These `lwarp-pagename.sty` files contain code used to emulate or replace functions for HTML output.

Enough functionality is provided to convert a typical article containing technical content. Not every package has been tested, but many of the most useful ones are known to work, either as-is or through emulation with replacement code. (See table 1 on page 34.)

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

2.1 Supported packages and features

Supported classes include book, report, and article. memoir is planned, but in the meantime many of the packages used by memoir are already supported.

Table 1 lists some of the various \TeX features which may be used. *Supported* means that the package or macro may be used as-is, perhaps with minor limitations. *Emulated* means that the original package or macro is not used, but similar functionality is provided in a way which is intended to be compatible with the user's \TeX code.

Table 1: \TeX -HTML generation — lwarp package — Supported functions

Category	Status
Engines:	<code>pdf\TeX</code> , <code>X\TeX</code> , <code>Lua\TeX</code>
Classes:	book, report, or article. memoir is planned.
Page layout:	Emulates geometry, fancyhdr, titleps, addlines, any-size, a4, a4wide, a5comb, textarea, zwpagelayout, typearea, ebook.

lwarf Supported Functions — continued

Category	Status
Sectioning:	Supported, with hyperlinks. Honors <code>tocdepth</code> and <code>secnumdepth</code> . Adds <code>FileDepth</code> for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation is removed from the filenames. Emulates <code>titlesec</code> , <code>fncychap</code> .
Table of contents, figures, tables:	Supported, with hyperlinks. Supports <code>tocbibind</code> . Emulates <code>titletoc</code> and <code>tocloft</code> .
Title page:	<code>\maketitle</code> , <code>titlepage</code> , <code>titling</code> , <code>authblk</code> .
Front & back matter:	<code>abstract</code> , <code>appendix</code>
Indexing:	<code>texindy</code> is used, with hyperlinks. <code>idxlayout</code> is emulated.
Glossary:	<code>glossaries</code> and <code>xindy</code> are used.
Bibliography:	Supported text-only.
Cross-references:	Emulated, with hyperlinks. <code>hyperref</code> , <code>cleveref</code> , <code>varioref</code> , <code>url</code> .
Languages:	<code>babel</code> . (<code>polyglossia</code> is untested.)
Margin notes:	<code>marginfit</code> , <code>marginfix</code> .
Footnotes:	<code>footnote</code> , <code>footmisc</code> , <code>marginnote</code> , <code>sidenote</code> , <code>pagenote</code> , <code>endnotes</code> .
Math:	Supported. Converted to SVG images with <code>HTML</code> ALT tags containing the <code>\TeX</code> source for the math expression. MathJax supported as an alternative. <code>\MS</code> environments are supported. User-defined macros are available during conversion, due to native <code>\TeX</code> processing.
Theorems:	Support for native <code>\TeX</code> theorems, plus <code>theorem</code> , <code>amsthm</code> , <code>ntheorem</code> .
Add'l math:	<code>delarray</code> , <code>bm</code> , math fonts via SVG images.
Units and fractions:	<code>siunitx</code> , <code>xfrac</code> , <code>nicefrac</code> , <code>units</code>

lwarf Supported Functions — continued

Category	Status
Floats:	Appear where declared. <code>float</code> , <code>rotfloat</code> , <code>newfloat</code> , <code>caption</code> and <code>subcaption</code> , <code>subfig</code> , <code>capt-of</code> , <code>placeins</code> , <code>trivfloat</code> , <code>floatrow</code> , <code>keyfloat</code> , <code>wrapfig</code> , <code>cutwin</code> , <code>floatfl</code> .
Tabular	<code>tabular</code> environment, <code>array</code> , <code>tabularx</code> , <code>tabulary</code> , <code>threeparttable</code> , <code>multirow</code> , <code>longtable</code> , <code>supertabular</code> , <code>xtab</code> , <code>booktabs</code> .
Graphics	<code>graphics</code> and <code>graphicx</code> are emulated. <code>\includegraphics</code> supports <code>width</code> , <code>height</code> , <code>origin</code> , <code>angle</code> , and <code>scale</code> tags, and adds <code>class</code> . References to PDF files are changed to SVG, other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle. <code>grffile</code> is emulated.
rotating:	Emulated. All objects are displayed unrotated.
picture and tikz:	Converted to an SVG image.
xcolor:	Supported. Full package color names, any color models, and <code>mixing</code> . <code>\textcolor</code> , <code>\colorbox</code> , <code>\fcolorbox</code> . Enhanced for HTML compatibility.
Lists:	Standard L ^A T _E X environments are supported, along with <code>enumitem</code> , <code>enumerate</code> , <code>paralist</code> . Spacing is still controlled by CSS.
Environments:	Standard L ^A T _E X environments are supported.
minipage:	Supported with some HTML-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
Quotations:	<code>verse</code> , <code>csquotes</code> , <code>epigraph</code>
Verbatim:	<code>verbatim</code> , <code>moreverb</code> , <code>fancyvrb</code> (except for verbatim footnotes).
Frames:	<code>framed</code> , <code>fancybox</code> , <code>mdframed</code> , <code>boxedminipage2e</code> , <code>shadow</code> .
multicol:	Emulated, with CSS. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.

l warp Supported Functions — continued

Category	Status
fullwidth:	Emulated.
Direct formatting:	\emph, \textsuperscript, \textbf, etc are supported. \bfseries, etc. are not yet supported. letter, ulem, soul, relsize, scalefnt, and realscripts are supported.
Ordinals:	nth, fmtcount, and engord are supported.
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 math expressions or embedded image environments.
Symbols:	Native L ^A T _E X diacriticals, textcomp.
Working as-is:	Various utility, calculation, file, and text-only packages, such as calc, fileerr, somedefs, trace, xspace.
Where:	
Supported:	The existing L ^A T _E X package is used.
Emulated:	The L ^A T _E X package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

3 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, lwarf 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.

3.1 Internet class

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

3.2 TeX4ht

Prog **TeX4ht** <http://tug.org/tex4ht/>

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, lwarf provides a better HTML conversion, and it supports a different set of packages. TeX4ht produces several other forms of output beyond HTML.

3.3 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** **H^Ev^Ea:** <http://hevea.inria.fr/> (not on CTAN)

Prog **TtH** **T_TH:** <http://hutchinson.belmont.ma.us/tth/>

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

Prog **LaTeXML** **LT_EXML:** <http://dlmf.nist.gov/LaTeXML/>

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

Prog **LaTeX2HTML** **LT_EX2HTML:** <http://www.latex2html.org/>
and <http://ctan.org/pkg/latex2html>.

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

Finally, Glad \TeX may used to directly insert \TeX math into HTML:

Prog Glad \TeX **Glad \TeX :** <http://humenda.github.io/GladTeX/>

3.4 AsciiDoc

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 \TeX and HTML.

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

Prog AsciiDoctor **AsciiDoc:** <http://asciidoc.org/> (The original version.)

The Asciidoctor-LaTeX project is adding additional \TeX -related features.

Asciidoctor-Latex:

<http://www.noteshare.io/book/asciidoctor-latex-manual>

Prog Asciidoctor-LaTeX <https://github.com/asciidoctor/asciidoctor-latex>

3.5 Pandoc

Prog Pandoc

A markup system which also reads and writes \TeX and HTML.

Pandoc: <http://pandoc.org/>

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

3.6 Word processors

Prog Word It should be noted that the popular word processors have advanced through the

Prog LibreOffice years in their abilities to represent math with a \TeX -ish input syntax, unicode math
Prog OpenOffice fonts, and high-quality output, and also generate HTML with varying success. See
recent developments in Microsoft® Word® and LibreOffice™ Writer.

3.7 Commercial systems

Prog Adobe 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™.

Prog InDesign

Prog Flare

Prog Madcap

3.8 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, L^AT_EX 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 L^AT_EX is comparably easy to learn, while L^AT_EX 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 L^AT_EX 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 L^AT_EX back end, yielding high-quality results especially when the L^AT_EX template is adjusted, but they lose the ability to use L^AT_EX macros and other L^AT_EX source-document features.

The effort required to customize the output of each markup system varies. For print output, L^AT_EX 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 lwarf, CSS is used, and much HTML output is adjusted through the usual L^AT_EX optional macro parameters, but further customization may require patching L^AT_EX 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.

4 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.

4.1 Installing the lwarf package

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

Pre-installed: Try entering into a command line:

Enter ⇒ kpsewhich lwarf.sty

If a path to lwarf.sty is shown, then lwarf is already installed.

T_EX Live: If using a T_EX Live distribution, try installing via tlmgr:

Enter ⇒ tlmgr install lwarf

MiK_TE_X: If using MiK_TE_X, try using the package installer to install the package lwarf. Also update the package miketex-misc, which will install the lwarpmk executable.

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

CTAN TDS archive: lwarf may be downloaded from the Comprehensive T_EX Archive:

1. See <http://ctan.org/pkg/lwarf> for the lwarf package.
2. Download the TDS archive: lwarf.tds.zip
3. Find the T_EX local directory:

T_EX Live:

Enter ⇒ kpsewhich -var-value TEXMFLOCAL

MiK_TE_X:

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

Table 2: Required software programs

Provided by your \TeX distribution:

From T_EXLive: <http://tug.org/texlive/>.

\TeX : pdflatex, xelatex, or lualatex.

The `lwarf` 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 lwarf 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 package is used by lwarf to create indexes. On a MiK_TE_X 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 \TeX code. On a MiK_TE_X system, Perl may need to be installed first.

pdfcrop: Used to pull images out of the \TeX PDF.

Poppler PDF utilities:

pdftotext: Used to convert PDF to text.

pdfseparate: Used to pull images out of the \TeX 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.

perl.org, strawberryperl.com

Automatically downloaded from the internet as required:

MathJax: Optionally used to display math. Automatically loaded from the MathJax website when needed.

From: mathjax.org

Or, for Windows MiK_TE_X, start the program called MiK_TE_X Settings (Admin) and click on the button called Refresh FNDB.

CTAN .dtx and .ins files: Another form of T_EX package is the .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 l warp package.

2. Download the zip archive l warp.zip into your own l warp directory.

3. Unpack l warp.zip.

4. Locate the contents l warp.dtx and l warp.ins

5. Create the documentation:

Enter ⇒ pdflatex l warp.dtx

(several times)

6. Create the .sty files:

Enter ⇒ pdflatex l warp.ins

7. Copy the .sty files somewhere such as the T_EX Live local tree found in the previous CTAN TDS section, under the subdirectory:

<texlocal>/tex/latex/local/l warp

8. Copy the documentation l warp.pdf to a source directory in the local tree, such as:

<texlocal>/doc/local/l warp

9. Renew the cache:

Enter ⇒ mktexlsr

— or —

Enter ⇒ texhash

Or, for Windows MiK_TE_X, start the program called MiK_TE_X Settings (Admin) and click on the button called Refresh FNDB.

10. See section 4.2.1 to generate your local copy of l warpmk.

11. Once the local version of l warpmk.lua is installed, it may be made available system-wide as per section 4.2.

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 l warpmk.lua files may be used as-is, so long as they are in the same directory as the document source. This approach is especially useful if you would like to temporarily test l warp before deciding whether to permanently install it.

4.2 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 [4.2.1](#).)

After the `lwarf` 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 [4.2.1](#).
2. For MiK \TeX , 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 MiK \TeX system you may use

Enter \Rightarrow `kpsewhich lwarpmk.lua`

(If the file is not found, you may also generate a local copy and use it instead. See section [4.2.1](#).)

4. Create `lwarpmk`:

Unix: Create a symbolic link and make it executable:

- (a) Locate the \TeX Live binaries:

Enter \Rightarrow `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 \Rightarrow `ln -s <path to lwarpmk.lua> lwarpmk`

- (c) Make the link executable:

Enter \Rightarrow `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 MiK \TeX : 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 l warpmk.bat containing:

```
texlua "C:\Program Files\MiKTeX 2.9\scripts\l warp\l warpmk.texlua" %*
```

This will call the copy of l warpmk.lua which is in the scripts directory of the distribution.

4.2.1 Using a local copy of l warpmk

It is also possible to use a local version of l warpmk:

1. When compiling the tutorial in section 5, use the l warpmk option for the l warp package:

```
\usepackage[l warpmk]{l warp}
```

2. When the tutorial is compiled with pdflatex, the file l warpmk.lua will be generated along with the other configuration files.

3. l warpmk.lua may be used for this project:

Unix:

- (a) Make l warpmk.lua executable:

Enter ⇒ chmod 0755 l warpmk.lua

- (b) Compile documents with

Enter ⇒ ./l warpmk.lua html

Enter ⇒ ./l warpmk.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 l warpmk.lua html

Enter ⇒ texlua l warpmk.lua print

etc.

Or:

Enter ⇒ l warpmk.html

Enter ⇒ l warpmk.print

etc.

4.3 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 42.

```
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 provided by your operating system's package manager, and are also provided by the CTAN archive:

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

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

Prog pdfseparate The tools from the POPPLER project should be provided by your operating system's package manager.
Prog pdftocairo

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 42.
2. Download and extract the Poppler utilities pdftotext, pdfseparate, and pdftocairo 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 Strawberry 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.

5 Tutorial

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

5.1 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:

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.)
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, L^AT_EX program (`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.

% --- 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
\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{l warp_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 L^AT_EX. 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 L^AT_EX 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: lwarf.css, lwarf_formal.css, lwarf_sagebrush.css These files are standard for lwarf, 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 lwarf defaults.

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

lwarf_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 lwarf to conditionally process blocks of text. This file may be ignored.

When the lwarpmk option is given to the lwarf 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.

5.2 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.³

5. Process the index.⁴⁵

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.

³Although, when using the utility `latexmk` (introduced later), the changed date is ignored and an actual change in contents must occur to cause a recompile.

⁴A `lwarpmk printglossary` command is also available to process a glossary produced with the `glossaries` package. See section 7.4.7.

⁵Also see section 7.4.8 for index options.

5.3 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`⁶, `Some-math.html`, and the document's index in `_Index.html`.⁷

2. View the homepage in a web browser.

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

math

Note that math is still displayed as its 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 5.4 and 5.5.

3. Force a recompile:

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk html`

Enter ⇒ `lwarpmk print`

4. Process the HTML index and recompile.⁸⁹

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`

⁶`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`.

⁷`index.html` is commonly used as a homepage, so the document index is in `_Index.html`.

⁸A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 7.4.7.

⁹Also see section 7.4.8 for index options.

5.4 Generating the SVG images

math as svg images

By default lwarf represents math as svg images with the \LaTeX source included in `alt` attributes. In this way, the math displays as it was drawn by \LaTeX , and the \LaTeX source may be copied and pasted into some other document.

picture and Tikz

lwarf uses the same mechanism for `picture` and `Tikz` environments.

1. Create the svg images:

Enter \Rightarrow `lwarfpmk limages`

Enter \Rightarrow `lwarfpmk html`

2. Move to the tutorial's math page and reload.
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 `Tikz` environment is added or removed, the svg images must be re-created with `lwarfpmk limages` to maintain the proper image file sequence numbers.

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.

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

Lots of files!

math expression. There is no attempt at reusing the same file each time the same expression occurs, so each time `x` is used, for example, yet another file is created. For a document with a large amount of math, see section 5.5 to use MathJax instead.

5.5 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.

5.6 Changing the CSS style

`\CSSFilename` `\CSSFilename` may be used to choose which .css file is used to display each section of the web page. 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.

The styles provided by lwarf include:

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

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

`lwarf_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 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 lwarf 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 `lwarf.css`, and this entry may be changed to load `lwarf_formal.css` or `lwarf_sagebrush.css` if desired. Additional changes to the css may be made by making entries later in the `<project>.css` file.

5.7 Customizing the HTML output

Several settings may be used to customize the HTML output. Watch for the correct placement of each!

⚠ Placement! 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`

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

`HomeHTMLFilename=index`

filename underscores

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:

`\item [\href{file_name.pdf}{text}] \`

Opt `HTMLFilename`

HTMLFilename: A filename prefix for the rest of the HTML web pages. Useful for numbered web pages with a common prefix. May be empty.

Opt `latexmk`

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

Opt `mathsvg`

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

Opt `mathjax`

mathjax: Selects MathJax for math output.

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

Ctr `tocdepth`

tocdepth: Sectioning depth of the table of contents. See section 13 for a list of `LATEX` stack depths.

Ctr **SideTOCDepth** **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`



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

Ctr **FileDepth** **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**

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. 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.

Inaccesible pages!

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 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.

HTML filenames Example HTML filenames:

Numbered HTML nodes:

Example: Homepage index.html, and node-1, node-2.¹⁰

```
\usepackage[
    HomeHTMLFilename=index,
    HTMLFilename={node-}
]{lwarf}
\boolfalse{FileSectionNames}
```

Named HTML sections, no prefix:

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

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

Named HTML sections, with prefix:

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

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

\abstractname \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 \CSSFilename: {<filename.css>} Sets the css file to use for the following files. May be changed before each sectioning command which would cause a file split.

¹⁰See \SetHTMLFileName to number grouped by chapter, for example.

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 L^AT_EX 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 pages of the website.

\HTMLLanguage	\HTMLLanguage: The HTML file's html lang tag. Defaults to en-US.
\HTMLAuthor	\HTMLAuthor: The HTML header's meta author. Defaults to \theauthor.
\HTMLDescription	\HTMLDescription: {\<description>} Sets the HTML description tag for the following files. May be changed before each sectioning command which would cause a file split.
\HTMLFirstPageTop	\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	\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	\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:

Env warpprint	warpprint: An environment which is only used while generating print output. Place here 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.
---------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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.
\warpprintonly	\warpprintonly: { <i>contents</i> } A macro version of the warpprint environment.
\warpHTMLonly	\warpHTMLonly: { <i>contents</i> } A macro version of the warpHTML environment.

5.8 Using `latexmk`

`latexmk` is a L^AT_EX 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.

[forced single-pass recompile](#) A single recompile may be forced with:

Enter ⇒ `lwarpmk print1`

and/or

Enter ⇒ `lwarpmk html1`

5.9 Using XeLaTeX or LuaLaTeX

Xe \TeX or Lua \TeX may be used instead of \TeX .

1. Remove the auxiliary files for the project:

Enter \Rightarrow lwarfmk cleanall

2. Use xelatex or lualatex to recompile the printed version.

Enter \Rightarrow xelatex tutorial.tex

-or-

Enter \Rightarrow lualatex tutorial.tex

When the recompile occurs, the configuration files for lwarfmk are modified to remember which \TeX engine was used. Xe \TeX or Lua \TeX will be used for future runs of lwarfmk.

3. To recompile the document:

Enter \Rightarrow lwarfmk print

-and-

Enter \Rightarrow lwarfmk html

4. Also rememeber to update the indexes and recompile again.

5.10 Using a glossary

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

Opt `IndexLanguage` To assign a language to be used while processing the index and glossary, use the `IndexLanguage` option:

```
\usepackage[IndexLanguage=english]{lwarf}
```

To process the glossary for the print version:

Enter ⇒ `lwarfmk printglossary`

To process the glossary for the HTML version:

Enter ⇒ `lwarfmk htmlglossary`

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

5.11 Cleaning auxiliary files

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

Enter ⇒ lwarpmk clean

5.12 Cleaning auxiliary and output files

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

Enter ⇒ lwarpmk cleanall

5.13 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 project_a

Enter ⇒ lwarpmk html project_b

5.14 Using the make utility

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

lwarpmk pdftohtml [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.

6 Additional details

6.1 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 bit-mapped fonts.

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



X_ET_EX and Lua_ET_EX users must use the fontspec package. Do NOT use fontenc!

Place fontspec or fontenc and other font and UTF-8 related commands after the \documentclass command and before \usepackage{lwarp}:

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

- For X_ET_EX or Lua_ET_EX:

Pkg **fontspec**

- fontspec and font choices

ligatures

lwarp sets the following to turn off T_EX 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:

Pkg **lmodern**

- lmodern or other font-related packages

Pkg **fontenc**

- fontenc

Pkg **inputenc**

- inputenc

Pkg **newunicodechar**

- newunicodechar

File	<code>glyptounicode</code>	- \input glyptounicode.tex - \input glyptounicode-cmr.tex% from the pdfx package - \pdfgentounicode=1
Pkg	<code>cmap</code>	- cmap
Pkg	<code>textcomp</code>	- textcomp
Pkg	<code>microtype</code> <code>ligatures</code>	- microtype is automatically used by lwarf to turn off f,q,t,TQ ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off \TeX ligatures.

3. `\usepackage{lwarf}` (section 6.2) goes after any of the above, followed by:
4. ... the rest of the preamble and the main document.

6.1.1 Indexes and UTF-8

lwarf uses the `xindy` program to processes indexes.

While using `xelatex` or `lualatex`, `xindy` is used for the index. Everything is handled in UTF-8 encoding, and should work as expected.

While using `pdflatex`, the `texindy` program is used with the `-C utf8` option, which is newly supported in recent distributions of \TeX . This option correctly sorts index entries into headings while using Latin languages, but will not work well with others. `X\TeX` or `Lua\TeX` are recommended for non-Latin languages.

For an older distribution of \TeX , it may be necessary to generate a local version of `lwarpmk.lua` and modify it to remove the `-C utf8` option from the `texindy` call. See section 10.4.

6.2 lwarf package loading and options

lwarf supports `book`, `report`, and `article` classes.

Pkg	<code>lwarf</code>	Load the lwarf package immediately after the font and UTF-8 setup commands.
Opt	<code>warpprint</code>	Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When lwarf is loaded in print mode, it creates <code><project>.html.tex</code> , which sets the <code>warpHTML</code> option before calling the user's source code <code><project>.tex</code> . In this way, <code><project>.tex</code> can <code>\usepackage{lwarf}</code> without any options to create a printed version, while <code><project>.html.tex</code> will create an HTML version.
Opt	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about
Opt	<code>mathjax</code>	

Table 4: Package options

Option	Description
<code>warpprint</code>	Generate print output, and also generate configuration files.
<code>warpHTML</code>	Generate HTML output.
<code>mathsvg</code>	Show math using SVG images.
<code>mathjax</code>	Show math using MathJax.
<code>OSWindows</code>	Force compatibility with MS-Windows.
<code>BaseJobname</code>	The <code>\jobname</code> to use. Set to the <code>\jobname</code> of the printed version even while generating HTML.
<code>HomeHTMLFilename</code>	The filename of the home page.
<code>HTMLFilename</code>	A prefix for the filenames of the remaining web pages.
<code>IndexLanguage</code>	The <code>xindy</code> language option used for index and glossary generation.
<code>latexmk</code>	Boolean for <code>lwarpmk</code> to use <code>latexmk</code> for compiling documents. Otherwise, <code>lwarpmk</code> attempts to recompile several times by itself.
<code>lwarpmk</code>	Generate a local copy of <code>lwarpmk.lua</code> .
<code>xdyFilename</code>	Tells <code>lwarpmk</code> to use a custom filename for <code>xindy</code> , instead of <code>lwarf.xdy</code> .

the math options, see section 7.5.

See table 4 for the full list of options.

6.3 Selecting the operating system

Prog Unix	<code>lwarf</code> 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 Mac OS	
Prog Linux	
Prog MS-Windows	If WINDOWS is not correctly detected, use the <code>lwarf</code> option <code>OSWindows</code> .
Prog Windows	
Opt OSWindows	When detected or specified, the operating-system path separator used by <code>lwarf</code> is modified, the boolean <code>usingOSWindows</code> is set true. This boolean may be tested by the user for later use.

6.4 Selecting actions for print or HTML output

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

For most of built-in L^AT_EX 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 HTML output is surrounded by a `warpHTML` environment:

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

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}
```

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}
```

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.

6.5 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 11: [Troubleshooting](#).

6.6 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, and the rest are used in \maketitle. To use a \subtitle or \published field, see section 52.7.

\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 52.7). In HTML, the title will appear in a

heading <h1>.

\author {\langle author \rangle}

 In \author, use \protect before formatting commands such as \textsc. In HTML, the author will appear in a <div> class author. \affiliation is a new addition to \warp.

\date {\langle date \rangle}

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

\thanks {\langle text \rangle}

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

6.7 HTML page meta descriptions

\HTMLDescription {\langle A description of the web page. \rangle} 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 description 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{}

6.8 HTML page meta author

\HTMLAuthor {\langle author \rangle} 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.

6.9 CSS

File `lwarp.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`. `project.css` should refer to `lwarp.css` as follows:

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

/* Load default lwarp settings: */
@import url("lwarp.css") ;
/* or lwarp_formal.css, lwarp_sagebrush.css */

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

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

An example file called `sample_project.css` is provided, and may be renamed `project.css`.

\CSSFilename For each section at which HTML files are split, \CSSFilename may be used before the sectioning command to select a css file for that and all following sections. This may be changed numerous times throughout the file, resulting in different HTML pages having different css files assigned:

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

6.10 Modifying xindy index processing

Prog `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 customize index processing:

1. Copy `lwarp.xdy` to a new filename such as `projectname.xdy`
2. Make changes to `projectname.xdy`. Keep the line which says

```
(markup-locref :open "\hyperindexref{" :close "}")
```

This line creates the hyperlinks for the HTML index. During print output `\hyperindexref` becomes a null function.

`Opt xdyFilename` 3. In the document source use the `xdyFilename` option for `lwarp`:

```
\usepackage[
    ... other options ...
    xdyFilename=projectname.xdy,
]{lwarp}
```

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`.

7 Special cases and limitations

Also see section 11: [Troubleshooting](#).

Some commonly-used L^AT_EX expressions should be modified to allow for a smooth conversion to both HTML and print-formatted outputs:

7.1 Formatting

7.1.1 Text formatting

`\textbf`, etc. are supported, but `\bfseries`, etc. are not yet supported.

7.1.2 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` `\kern` and `\hskip` are treated as a single normal space.
`\hskip`

7.1.3 Text alignment

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

7.1.4 Accents

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

7.1.5 Textcomp

-  **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.

7.1.6 Superscripts and other non-math uses of math mode

Use `x` instead of `x`

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

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

7.1.8 Filenames in lists

- filename underscore** Escape underscores in the filenames:

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

7.1.9 relsize package

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 temporarily restored, and so should work as expected.

-  **not small** The `HTML` browser's setting for minimum font size may limit how small the output will be displayed.

7.2 Boxes and minipages

7.2.1 Save Boxes

\TeX 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.

7.2.2 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`, `\quad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print and HTML . Paragraph tags are suppressed 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 ``. An HTML `<div>` cannot appear inside a ``. While in a ``, 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 \TeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

7.2.3 Side-by-side minipages

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

7.2.4 Framed minipages and other environments

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

	<code>\fbox</code>	For a framed object, options include:
	<code>\fboxBlock</code>	
Env	<code>fminipage</code>	To remove the frame in HTML output: Place the <code>\fbox</code> command and its closing brace inside <code>\warpprint</code> environments. This will nullify the frame for HTML output.
		To frame the contents inline with some formatting losses in HTML: This is the default action of <code>\fbox</code> when enclosing a minipage. During HTML output, <code>\fbox</code> nullifies the HTML tags for <code>minipage</code> , <code>\parbox</code> , and lists. The contents are included as inline text inside the <code>\fbox</code> 's <code></code> of class <code>framebox</code> . 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 text:		To frame the contents on their own line with improved formatting in HTML: A new command <code>\fboxBlock</code> is included, intended to be a direct replacement for <code>\fbox</code> for cases where the <code>\fbox</code> surrounds a minipage, table, or list. For print output, this behaves as <code>\fbox</code> . For HTML output, the contents are placed inside an HTML <code><div></code> with the class <code>framed</code> , 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 <code>tabular</code> . While an <code>\fbox</code> containing a <code>tabular</code> is valid L ^A T _E X code, the result in HTML is problematic since a table is a <code><div></code> not a <code></code> , so use <code>\fboxBlock</code> around a <code>tabular</code> , or else place the <code>tabular</code> inside a <code>minipage</code> , or use <code>fminipage</code> , described next. Also see below regarding the “Misplaced alignment tab character &” error.
For inline <code>minipage</code> and lists:		To create a framed minipage in both print and HTML: A new environment <code>fminipage</code> is included. For print output, this is identical to <code>minipage</code> , except that it is also framed. For HTML output, this forms a <code><div></code> of class <code>framed</code> , the contents preserve their HTML formatting, and valid HTML is created for a <code>tabular</code> . Also see below regarding the “Misplaced alignment tab character &” error.
For display <code>tabular</code> , <code>minipages</code> , and lists:		

colored boxes and frames: **To create colored frames and boxes:** See section 228 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 \EndDefiningTabulars afterwards. Also see the lwarp documentation for the fancybox package.

To frame equations: See section 119 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}
```

7.2.5 fancybox package

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.

```
\newenvironment{FramedEqn}
{
    \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}
\begin{Verbatim}
\end{Verbatim}
\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.

7.2.6 mdframed package

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 lwarf 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.

7.3 Cross-references

`\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.

7.3.1 Page references

The printed page does not translate to the HTML page, so references to page numbers are converted to parentheses containing `\pageref{PageFor}`, which defaults to “see”, followed by a hyperlink to the appropriate object. Ex: “Sec. 1.23 on page (see sec. 1.23)”. `\pageref{PageFor}` may be redefined to “page for”, empty, etc.

7.3.2 cleveref and varioref packages

`cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used instead. See section 7 to redefine the message which is printed for page number references.

7.3.3 Hyperlinks, hyperref, and url

lwarf emulates `hyperref`, including the creation of active hyperlinks, but does not require that `hyperref` be loaded by the document.

lwarf can also load `url`, but `url` should not be used at the same time as `hyperref`, since they both define the `\url` command. lwarf does not (yet) attempt to convert `url`

links into hyperlinks during HTML output, nor does url create hyperlinks during print output.

7.3.4 Footnotes and page notes

lwarf uses native L^AT_EX footnote code, although with its own \box to avoid the L^AT_EX output routine. The usual functions work as-is.

7.4 Front and back matter

7.4.1 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 lwarf.

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.

lwarf 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*. lwarf 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. 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.



\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*. The TOC will only be assigned for HTML output, not for print output, and it will appear in the main TOC and also the sidetoc per page.

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

7.4.2 abstract package

 **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.

7.4.3 titling and authblk

package support lwarf supports the native L^AT_EX 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 52.7.

7.4.4 appendix package

 **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.

7.4.5 pagenote package

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

7.4.6 endnotes package

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
```

7.4.7 glossaries package

Pkg **glossaries** xindy is required for glossaries.

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

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

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 HTMLpage:
----------------------------------	----------------------------------------------------------------------------------------------------------------------------------

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
```

Opt **IndexLanguage** The lwarf package takes an option IndexLanguage=english to set the language used by xindy. This is passed to xindy using its -L option, and is used for both index and glossary generation.

Opt **lwarfmk printglossary** lwarfmk has the commands lwarfmk printglossary and lwarfmk htmlglossary

Opt **lwarfmk htmlglossary** to process the glossaries created by glossaries using xindy.

7.4.8 Index and the **tocbibind** package

Opt **IndexLanguage** The lwarf package takes an option IndexLanguage=english to set the language used by xindy. This is passed to xindy using its -L option, and is used for both index and glossary generation.

placement and toc options	An index may be placed inline with other HTML text, or on its own HTML page:
----------------------------------	------------------------------------------------------------------------------

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
```

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
```

Inline, with an automatic TOC entry:

Pkg tocbibind

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
```

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.

See section 62.

7.5 Math

 **Equation numbering**

ntheorem has a bug with equation numbering in \mathcal{AM} S environments when the option `thref` is used. lwarf does not share this bug, so equations with `\split`, etc, are numbered correctly with lwarf's HTML output, but not with the print output. It is recommended to use `cleveref` instead of ntheorem's `thref` option.

Math rendering	Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.
SVG math option	For SVG math, math is rendered as usual by L ^A T _E X into the initial PDF file using the current font ¹¹ , 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 L ^A T _E X 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 L ^A T _E X code which generated the math, allowing copy/paste of the L ^A T _E X math expression into other documents.
SVG image font size	The size of the math and text used in the SVG image may be adjusted by setting \LateximageFontSizeName to a font size name — <i>without the backslash</i> , for ex: <code>\renewcommand{\LateximageFontSizeName}{large}</code>
SVG files	As currently implemented, each instance of math creates a new SVG file. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.
SVG inline	Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images, but dis-allows the possibility of sharing one file among many instances without user intervention.
PNG files	Others 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 also larger as well.
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 8 regarding EPUB output with MathJax.
MathJax math option	<p>The popular MathJax alternative (mathjax.org) may be used to display math.</p> <p>Prog MathJax</p> <p>When MathJax is enabled, math is rendered twice:</p> <ol style="list-style-type: none"> 1. As regular L^AT_EX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of L^AT_EX, and 2. As detokenized printed L^AT_EX 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 L^AT_EX values, and the MathJax cross-referencing system is ignored in favor of the L^AT_EX internal system, seamlessly integrating with the rest of the L^AT_EX code.

¹¹See section 229 regarding fonts and fractions.

MathJax limitations

Prog MathJax

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 Tikz 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 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/mathjax/MathJax-third-party-extensions/tree/master/siunitx>

 \LaTeX macros

- MathJax does not automatically support custom \LaTeX macros, but they may be set up by the user.

custom MathJax macros

For an example of using custom \LaTeX macros with MathJax, see page 321.

7.5.1 ntheorem package**⚠ 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.

7.5.2 siunitx package**Pkg siunitx**

Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

⚠ per-mode

Some units will require that the expression be placed inside math mode.

7.5.3 units and nicefrac packages

Pkg **units** units and nicefrac work as-is with lwarp, but MathJax does not have an extension for units or nicefrac. These packages do work with lwarp's option `svgmath`.

7.5.4 newtxmath package

Pkg **newtxmath** The proper load order is:

⚠ **loading sequence**

1. ...
2. `\usepackage{lwarp}`
3. ...
4. `\usepackage{amsthm}`
5. `\usepackage{newtxmath}`
6. ...

7.6 Graphics

⚠ **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 are not supported by warp; the entire image will be shown.

⚠ **viewports**

⚠ **\graphicspath** `\graphicspath` only works for a single directory; all graphics must be in this directory.

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. Using 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.

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.

⚠ **image file types**

For `\includegraphics` the user should provide both `.pdf` and `.svg` images, but always refer to `.pdf` images in the document source. All `\includegraphics` references to `.pdf` will automatically be changed to `.svg` for `HTML` output, and will be left as `.pdf` for print output. Images may also be `.jpg` and `.png`, and will be used as-is for either output.

`\rotatebox` `\rotatebox` accepts the optional `origin` key.

⚠ **browser support**

`\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The `css` 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.

7.6.1 grffile package

`grffile` is partially emulated. 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` version for `HTML`. No attempt is made to convert input or filename encodings.

⚠ **matching PDF and SVG
encodings**

7.6.2 color package

`color` is superceded by `xcolor`, and `lwarf` 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.

7.6.3 xcolor package

`\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 ``. These `<div>`s are set to `display: inline-block` so adjacent `\colorboxBlocks` appear side-by-side in `HTML`, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by `lwarf`'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.

`tables` Colored tables are ignored so far. Use `css` to style tables.

colored text and boxes `\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

\color and \pagecolor `\color` and `\pagecolor` are ignored. Use `css` or `\textcolor` where possible.

7.7 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 `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 ignored in print mode.

`\StartDefiningTabulars`

`<define macros or environments using tabular and &
here>`

`\EndDefiningTabulars`

**⚠ tabular inside another
environment**

- When creating a new environment which contains a `tabular` environment, lwarf's emulation of the `tabular` does not automatically resume when the containing environment 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`

`}`

`\EndDefiningTabulars`

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.

- 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.

 \multicolumn & \\multirow

- lwarp does not support 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. See section 168.2.

- 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 types

 * column specification

- Vertical rules are not yet supported.
- * in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of @, !, >, and < may be used at each column, and they are used in that order.
- \newcolumntype is ignored; unknown column types are set to l.

Rules:

\multirow with rules

rule at last row

\cmidrule width, trim

\cmidrule and \\multicolumn

 \\warpprintonly misplaced \\noalign

- If a multirow reaches to the bottom of a table, and \bottomrule does not go over to that edge, try adding a line of empty cells below the \bottomrule. (This may be a browser bug.)
- If a \midrule is desired after the last row, an additional row of blank cells must be used.
- \cmidrule does not support width or trim options due to css limitations.
- \cmidrule borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has \multicolumn cells below \cmidrule borders.
- 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 relevant to HTML output.

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 \\}
```

7.7.1 longtable package

⚠ 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 relevant to HTML output.

⚠

\kill is ignored, place a \kill line inside

```
\begin{warpprint} ... \end{warpprint}
```

or place it inside \warpingprintonly.

7.7.2 supertabular and xtab packages

For \tablefirsthead, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

**⚠ misplaced alignment
alignment tab character &**

See section 7.7.

7.7.3 float, trivfloat, and/or algorithmicx together

-  **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 22.1.

7.7.4 caption and subcaption packages

To ensure proper float numbering, set caption positions such as:

```
\captionsetup[table]{position=top}
\captionsetup[figure]{position=bottom}
```

Similarly for `subtable`, `subfigure`, and `longtable`.

7.8 Floats

7.8.1 subfig package

-  **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.

7.8.2 floatrow package

-  **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. `lwarf` 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.

7.9 Miscellaneous

7.9.1 `verse` and `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. `lwarf` 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 \leftskip`
`Len \leftmargini`
`Len \TMLvleftskip`
`Len \TMLleftmargini` 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.

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.

7.9.2 `newclude` package

`Pkg newclude` `newclude` modifies `\label` in a non-adaptive way, so `newclude` must be loaded before `lwarf` is loaded.
⚠ loading

Ex:

```
\documentclass{article}
...<font setup>
\usepackage{newclude}
\usepackage[warpHTML]{lwarp}
...
...
```

7.9.3 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**

8 EPUB conversion

lwarf 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 lwarf 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 lwarf HTML output to EPUB:

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 lwarf HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

to the project's source preamble after \usepackage{lwarf}. The EPUB version of the document cannot co-exist with the regular HTML version, so

```
Enter ⇒ lwarfmk cleanall
```

then

```
Enter ⇒ lwarfmk html
```

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 file split.

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 lwarf 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

Check the box Add linked files in breadth first order.

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 5. 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.

9 Word-processor conversion

lwarf 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. lwarf provides some assistance in locating these frame boundaries, as shown below.

A special boolean is provided to simplify the process of converting lwarf HTML output to EPUB:

Bool FormatWordProcessor

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.

To help modify lwarf HTML output for easier import to a word processor, add

\booltrue{FormatWordProcessor}

to the project's source preamble after lwarf is loaded. 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 file split. These will have to be manually moved to their proper place in the document. lwarf does not know where the page breaks will be in the word processor's document, so the footnotes are simply moved to the end of each sectional break.
- 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 boolean may be set to help mark float boundaries:

Default true. Adds

```
    === table begin
    ...
    === end
Bool  HTMLMarkFloats  or
    === figure begin
    ...
    === end
```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames.

When enabled, markers are placed around each float, helping the user to identify float boundaries for further conversion to word processor frames and captions.

10 Modifying l warp

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.

An environment may be converted to a `\latextimage` then displayed with an image of the resulting `\TeX` output. See section 70 for an example of the `picture` environment.

To create a custom HTML block or inline css class, see section 39.8.

10.1 Creating an l warp version of a package

When creating HTML, l warp redefines the `\usepackage` and `\RequirePackage` macros such that it first looks to see if a `l warp-<packagename>.sty` version exists. If so, the l warp version is used instead. This modular system allows users to create their own versions of packages for l warp to use for HTML, simply by creating a new package with a `l warp-` 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 `l warp-` packages where `\TeX` can see it, then the user's new package will be seen by any documents using l warp. (Remember `mktexlsr` or `texhash`.)

An `l warp-<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 `l warp-` packages for examples.

Anything the user might expect of the original package must be replaced or emulated by the new `l warp-` 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 `l warp-` package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If Dropped, 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 Passed, the original package is loaded first, with the user-supplied options, then the lwarp-version continues loading as well. See section 177 (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.

10.2 Adding a package to the `lwarp.dtx` file

When adding a package to `lwarp.dtx` for permanent including 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 20. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`.

10.3 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.

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.

10.4 Modifying `lwarpmk`

Prog `lwarpmk` 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.
File `lwarpmk.lua`

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`.

To adjust the command-line arguments for compiling the document, look in `mymake.lua` for “`latexname`”.

To adjust the command-line arguments for processing the index, look for “`xindy`”.

11 Troubleshooting

11.1 Using the lwarf.sty package

Also see:

Section 6.5: Commands to be placed into the warpprint environment

Section 7: Special cases and limitations

Section 7: Special cases and limitations

Text is not converting:

- Font-related UTF-8 information must be embedded in the PDF file. See section 6.1 regarding vector fonts.

Undefined HTML settings:

- See the warning regarding the placement of the HTML settings at section 5.7.

Tabular problems: See section 7.7.

Obscure error messages:

- Be sure that a print version of the document compiles and that your document's L^AT_EX code is correct, before attempting to generate an HTML version.

Missing sections:

- See section 5.7 regarding the FileDepth and SideTOCDepth counters, and the use of \tableofcontents in the home page.

Missing HTML files:

See the warning regarding changes to the HTML settings at section 5.7.

Missing / incorrect cross-references:

- Use lwarfmk again followed by lwarfmk html or lwarfmk print to compile the document one more time.
- \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 instead. See section 7 to redefine the message which is printed for page number references.

Em-dashes or En-dashes in listing captions and titles:

Use X_ET_EX or Lua_ET_EX.

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.

Print document contains HTML tags:

- Be sure that the document selects \usepackage [warpprint] {lwarf} instead of [warpHTML].

HTML document contains a single unformatted print document:

- Be sure that the document selects \usepackage [warpHTML] {lwarf} instead of [warpprint].

Images are appearing in strange places:

- lwarpmk limages to refresh the lateximage images.

⚠️ Adding/removing

SVG images: When a math expression, picture, or Tikz environment is added or removed, the SVG images must be re-created with lwarpmk limages to maintain the proper image file sequence numbers.

⚠️ 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.

⚠️ Lots of files!

Expressing math as SVG images has the advantage of representing the math exactly as E_TT_EX would, but has the disadvantage of requiring an individual file for each math expression. There is no attempt at reusing the same file each time the same expression occurs, so each time \$x\$ is used, for example, yet another file is created. For a document with a large amount of math, see section 5.5 to use MathJax instead.

“Leaders not followed by proper glue”: This can be caused by a missing l@<floattype> or l@<sectiontype> definition. See lwarf’s definitions for examples.

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 [5.7](#).
- 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.

11.1.1 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.

11.2 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 into 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. `pdflatex lwarp.ins` to generate a new `lwarp.sty` file.
3. `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}`

12 Implementation

This package is perhaps best described as a large collection of smaller individual technical challenges, in many cases solved through a number of *erude-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 relevant 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 `\teximage` environment to preserve the graphics.

Testing has primarily been done with the Iceweasel/Firefox browser.

13 Stack depths

Stacks are created to track depth inside the `LATEX` document structure. This depth is translated to `HTML` headings as shown in table 5. “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 `HTML` `<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.

Table 5: Section depths and HTML headings

Section	\LaTeX depth	HTML headings
title of the entire website		<code><h1></code>
none	-5	new for this package
book	-2	not yet used
part	-1	<code><h2></code>
chapter	0	<code><h3></code>
section	1	<code><h4></code>
subsection	2	<code><h5></code>
subsubsection	3	<code><h6></code>
paragraph	4	<code></code>
subparagraph	5	<code></code>
listitem	7	new for this package, used for list items

14 Source Code

This is where the documented source code for lwarf 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 lwarf package.

line numbers The small numbers at the left end of a line refer to line numbers in the `lwarf.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 margin are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are

index entries 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”.

 **warnings** Special warnings are marked with a warning icon.

for HTML output: 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.
for PRINT output:
for HTML & PRINT:

15 Detecting the \TeX Engine — pdflatex, lualatex, xe-latex

```
1 \RequirePackage{iftex}
2
3 \ifLuaTeX
4 \RequirePackage{lualatex85}% until the geometry package is updated
5 \fi
```

16 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 `\textttimes`.

The first arguments of `\newunicodechar` below are text ligatures in the source code, even though they are not printed in the following listing.

```
6
7 \RequirePackage{newunicodechar}
8
9 \newunicodechar{x}{\textttimes}
10
11 \ifPDFTeX
12 \newunicodechar{ff}{ff}% the first arguments are ligatures
13 \newunicodechar{fi}{fi}
14 \newunicodechar{fl}{fl}
15 \newunicodechar{ffi}{ffi}
16 \newunicodechar{ffl}{ffl}
17 \newunicodechar{--}{--}
18 \newunicodechar{-}{--}
```

In PDFT \TeX , preserve upright quotes in verbatim text:

```
19 \RequirePackage{upquote}
20 \else
21 \fi
```

17 Early package requirements

Pkg `etoolbox` Provides `\ifbool` and other functions.

22 \RequirePackage{etoolbox}[2011/01/03]
23 % requires v2.6 for \BeforeBeginEnvironment, etc.

Pkg `ifplatform` Provides `\ifwindows` to try to automatically detect Windows OS.

24 \RequirePackage{ifplatform}%
sense op-system platform

Pkg `letltxmacro` Used to redefine `\textbf` and friends.

25 \RequirePackage{letltxmacro}

18 Operating-System portability

Prog Unix	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 Mac OS	
Prog Linux	
Prog MS-Windows	If WINDOWS is not correctly detected, use the lwarp option <code>OSWindows</code> .
Prog Windows	
Opt OSWindows	When detected or specified, the operating-system path separator used by lwarp is modified, the boolean <code>usingOSWindows</code> is set true. This boolean may be tested by the user for later use.

18.1 Common portability code

Bool `usingOSWindows` Set if the `OSWindows` option is used.

```
26 \newbool{usingOSWindows}
27 \boolfalse{usingOSWindows}
```

18.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

```
28 \newcommand*{\OSPathSymbol}{/}
```

18.3 MS-Windows

For MS-Windows:

\LWR@setOSWindows Set defaults for the MS-Windows operating system. lwarp attempts to auto-detect the operating system, and the `OSWindows` option may also be used to force MS-Windows compatibility.

```
29 \newcommand*{\LWR@setOSWindows}
30 {
31 \booltrue{usingOSWindows}
32 \renewcommand*{\OSPathSymbol}{\@backslashchar}
33 }
```

Test for windows during compile. The user may also specify OSWindows package option in case this test fails.

```
34 \ifwindows
35 \LWR@setOSWindows
36 \fi
```

19 Package options

Pkg kvoptions Allows key/value package options.

```
37 \RequirePackage{kvoptions}
38 \SetupKeyvalOptions{family=LWR,prefix=LWR@}
```

Bool warpingprint

Bool warpingHTML Set to true/false depending on the package option selections for print/HTML/EPUB output and mathsvg/mathjax:

```
39 \newbool{warpingprint}
40 \newbool{warpingHTML}
41 \newbool{mathjax}
```

defaults The default is print output, and svg math if the user chose HTML output.

```
42 \booltrue{warpingprint}%
43 \boolfalse{warpingHTML}%
44 \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.

```
45 \DeclareVoidOption{warpprint}{%
46 \PackageInfo{lwarp}{Using option 'warpprint'}
47 \booltrue{warpingprint}%
48 \boolfalse{warpingHTML}%
49 }
```

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.

```
50 \DeclareVoidOption{warpHTML}{%
51 \PackageInfo{lwarp}{Using option 'warpHTML'}%
```

```

52 \booltrue{warpingHTML}%
53 \boolfalse{warpingprint}%
54 }
```

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.

```

55 \DeclareVoidOption{mathsvg}{%
56 \PackageInfo{l warp}{Using option 'mathsvg'}%
57 \boolfalse{mathjax}%
58 }
```

Opt `mathjax` Option `mathjax` selects MathJax math display: If the `mathjax` option is given, boolean `mathjax` is true, may be used for `\ifbool` tests.

```

59 \DeclareVoidOption{mathjax}{%
60 \PackageInfo{l warp}{Using option 'mathjax'}%
61 \booltrue{mathjax}%
62 }
```

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.

```
63 \DeclareStringOption[\jobname]{BaseJobname}
```

Opt `IndexLanguage` Sets the language to be assigned in `lwarpmk`'s configuration files. This is then used by `lwarpmk` while processing the index and glossary.

```
64 \DeclareStringOption[english]{IndexLanguage}
```

Opt `xdyFilename` Selects a custom `.xdy` file. The default is `l warp.xdy`. A customized file should be based on `l warp.xdy`, and must retain the line

```
(markup-locref :open "\hyperindexref{" :close "})
```

```
65 \DeclareStringOption[l warp.xdy]{xdyFilename}
```

Opt `lwarpmk` Tells `l warp` 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`:

```

66 \newbool{LWR@creatinglwarpmk}
67 \boolfalse{LWR@creatinglwarpmk}
68
69 \DeclareVoidOption{lwarpmk}{
70 \PackageInfo{lwarp}{Using option 'lwarpmk'}
71 \booltrue{LWR@creatinglwarpmk}
72 }

```

- Opt `OSWindows` Tells `lwarp` to use MS-Windows compatibility. Auto-detection of the operating system 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.

```

73 \DeclareVoidOption{OSWindows}{
74 \PackageInfo{lwarp}{Using option 'OSWindows'}
75 \LWR@setOSWindows
76 }

```

- Opt `HomeHTMLFilename` 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.

```
77 \DeclareStringOption[] {HomeHTMLFilename}
```

- Opt `HTMLFilename` 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.

```
78 \DeclareStringOption[] {HTMLFilename}
```

- Opt `latexmk` Option `latexmk` tells `lwarpmk` to use `latexmk` when compiling documents.

```
79 \DeclareBoolOption[false]{latexmk}
```

- Execute options** Execute the package options, with the defaults which have been set just above:

```
80 \ProcessKeyvalOptions*\relax
```

Assign the `\BaseJobname` if the user hasn't provided one:

```
81 \providecommand*{\BaseJobname}{\LWR@BaseJobname}
```

Defaults unless already over-ridden by the user:

```
82 \ifcsempty{\LWR@HomeHTMLFilename}{
83 \newcommand*{\HomeHTMLFilename}{\BaseJobname}
```

```

84 }{
85 \csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
86 }
87
88 \csedef{HTMLFilename}{\LWR@HTMLFilename}

```

19.1 Conditional compilation

\warpprintonly {\langle contents \rangle}

Only process the contents if producing printed output.

```
89 \newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}
```

\warpHTMLonly {\langle contents \rangle}

Only process the contents if producing HTML output.

```
90 \newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}
```

Pkg comment Provides conditional code blocks.

```
91 \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.

```

92 \ifbool{warpingHTML}{
93 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
94 }{}
95
96 \ifbool{warpingprint}{
97 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
98 }{}

```

```
99 \excludecomment{testing}
```

Env warpall Anything in the `warpall` environment will be generated for print or HTML outputs.

```
100 \includecomment{warpall}
```

Env warpprint Anything in the `warpprint` environment will be generated for print output only.
 Env warpHTML

For HTML output:

```

101 \ifbool{warpingHTML}{%
102 \includecomment{warpHTML}
103 }
104 {\excludecomment{warpHTML}}%

105 \ifbool{warpingprint}
106 {\includecomment{warpprint}}
107 {\excludecomment{warpprint}}

```

Optionally generate a local copy of l warpmk. Default to no.

```

108 \ifbool{LWR@creatinglwarpmk}
109 {\includecomment{LWR@createlwarpmk}}
110 {\excludecomment{LWR@createlwarpmk}}

```

20 Misplaced packages

Several packages should only be loaded before l warp, and others should only be loaded after.

Packages which should only be loaded before l warp have their own

`l warp-<packagename>.sty`

which will trigger an error if they are loaded after l warp. Examples include fontspec, inputenc, fontenc, and newunicodechar.

`\LWR@loadafter {<packagename>}` Error if this package was loaded before l warp.

```

111 \newcommand*{\LWR@loadafter}[1]{%
112 \@ifpackageloaded{#1}{%
113 {%
114 \PackageError{l warp}{%
115 {Package #1, or one which uses #1, must be loaded after l warp}%
116 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{l warp}.}%
117 Package #1 may also be loaded by something else, which must also be moved%
118 after l warp.}%
119 }%
120 {}%
121 }

```

`\LWR@loadbefore {<packagename>}` Error if this package is after l warp.

```

122 \newcommand*{\LWR@loadbefore}[1]{%
123 \@ifpackageloaded{#1}%
124 {}%
125 {%
126 \PackageError{lwarp}%
127 {Package #1 must be loaded before lwarp}%
128 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}%
129 }%
130 }

```

\LWR@loadnever {\langle badpackagename\rangle} {\langle replacementpkgnname\rangle}

The first packages is not supported, so tell the user to use the second instead.

```

131 \newcommand*{\LWR@loadnever}[2]{%
132 \PackageError{lwarp}%
133 {Package #1 is not supported by lwarp's HTML conversion.%
134 Package(s) #2 may be useful instead}%
135 {Package #1 might conflict with lwarp in some way,%
136 or is superceded by another package.%
137 For a possible alternative, see package(s) #2.}%
138 }

```

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:

```

139 \LWR@loadafter{a4}%
140 \LWR@loadafter{a4wide}%
141 \LWR@loadafter{a5comb}%
142 \LWR@loadafter{abstract}%
143 \LWR@loadafter{addlines}%
144 \LWR@loadafter{afterpage}%
145 \LWR@loadafter{algorithmicx}%
146 \LWR@loadafter{alltt}%
147 \LWR@loadafter{amsthm}%
148 \LWR@loadafter{anyfontsize}%
149 \LWR@loadafter{appendix}%
150 \LWR@loadafter{arabicfront}%
151 \LWR@loadafter{array}%
152 \LWR@loadafter{authblk}%
153 \LWR@loadafter{balance}%
154 \LWR@loadafter{bigdelim}%
155 \LWR@loadafter{bigstrut}%
156 \LWR@loadafter{bookmark}%
157 \LWR@loadafter{booktabs}%
158 \LWR@loadafter{boxedminipage}%

```

```
159 \LWR@loadafter{boxedminipage2e}
160 \LWR@loadafter{ccaption}
161 \LWR@loadafter{changepage}
162 \LWR@loadafter{chngpage}
163 \LWR@loadafter{chappg}
164 \LWR@loadafter{color}
165 \LWR@loadafter{crop}
166 \LWR@loadafter{cutwin}
167 \LWR@loadafter{dcolumn}
168 \LWR@loadafter{draftwatermark}
169 \LWR@loadafter{ebook}
170 \LWR@loadafter{ellipsis}
171 \LWR@loadafter{emptypage}
172 \LWR@loadafter{enumerate}
173 \LWR@loadafter{enumitem}
174 \LWR@loadafter{epigraph}
175 \LWR@loadafter{eso-pic}
176 \LWR@loadafter{everypage}
177 \LWR@loadafter{everyshi}
178 \LWR@loadafter{extramarks}
179 \LWR@loadafter{fancybox}
180 \LWR@loadafter{fancyhdr}
181 \LWR@loadafter{fancyvrb}
182 \LWR@loadafter{figcaps}
183 \LWR@loadafter{fix2col}
184 \LWR@loadafter{float}
185 \LWR@loadafter{floatflt}
186 \LWR@loadafter{floatpag}
187 \LWR@loadafter{floatrow}
188 \LWR@loadafter{flushend}
189 \LWR@loadafter{fncychap}
190 \LWR@loadafter{ftnright}
191 \LWR@loadafter{fullpage}
192 \LWR@loadafter{fullwidth}
193 \LWR@loadafter{geometry}
194 \LWR@loadafter{glossaries}
195 % \LWR@loadafter{graphics}%
196 % \LWR@loadafter{graphicx}%
197 \LWR@loadafter{grffile}
198 \LWR@loadafter{hyperref}
199 \LWR@loadafter{hyperxmp}
200 \LWR@loadafter{idxlayout}
201 \LWR@loadafter{indentfirst}
202 \LWR@loadafter{keyfloat}
203 \LWR@loadafter{layout}
204 \LWR@loadafter{letterspace}
205 \LWR@loadafter{lettrine}
206 \LWR@loadafter{lips}
207 \LWR@loadafter{listings}
208 \LWR@loadafter{longtable}
```

```
209 \LWR@loadafter{lscape}
210 \LWR@loadafter{ltcaption}
211 \LWR@loadafter{marginfit}
212 \LWR@loadafter{marginfix}
213 \LWR@loadafter{marginnote}
214 \LWR@loadafter{mcaption}
215 \LWR@loadafter{mdframed}
216 \LWR@loadafter{metalogo}
217 \LWR@loadafter{microtype}
218 \LWR@loadafter{moreverb}
219 \LWR@loadafter{mparhack}
220 \% \LWR@loadafter{multicol} % loaded by ltxdoc
221 \LWR@loadafter{multirow}
222 \LWR@loadafter{nameref}
223 \LWR@loadafter{needspace}
224 \LWR@loadafter{newtxmath}
225 \LWR@loadafter{nextpage}
226 \LWR@loadafter{nonumonpart}
227 \LWR@loadafter{nopageno}
228 \LWR@loadafter{nowidow}
229 \LWR@loadafter{ntheorem}
230 \LWR@loadafter{pagenote}
231 \LWR@loadafter{paralist}
232 \LWR@loadafter{parskip}
233 \LWR@loadafter{pdflandscape}
234 \LWR@loadafter{pdfsync}
235 \LWR@loadafter{placeins}
236 \LWR@loadafter{prelim2e}
237 \LWR@loadafter{ragged2e}
238 \LWR@loadafter{realscripts}
239 \LWR@loadafter{relsize}
240 \LWR@loadafter{romanbar}
241 \LWR@loadafter{romanbarpagenumber}
242 \LWR@loadafter{rotating}
243 \LWR@loadafter{rotfloat}
244 \LWR@loadafter{savetrees}
245 \% \LWR@loadafter{scalefnt} % loaded by babel-french
246 \LWR@loadafter{setspace}
247 \LWR@loadafter{shadow}
248 \LWR@loadafter{showidx}
249 \LWR@loadafter{showkeys}
250 \LWR@loadafter{sidecap}
251 \LWR@loadafter{sidenotes}
252 \LWR@loadafter{siunitx}
253 \LWR@loadafter{soul}
254 \LWR@loadafter{subfig}
255 \LWR@loadafter{supertabular}
256 \LWR@loadafter{tabularx}
257 \LWR@loadafter{tabulary}
258 \LWR@loadafter{textarea}
```

```

259 % \LWR@loadaft{textcomp}%
260 \LWR@loadaft{textpos}
261 \LWR@loadaft{theorem}
262 \LWR@loadaft{threeparttable}
263 \LWR@loadaft{tikz}
264 \LWR@loadaft{titleps}
265 \LWR@loadaft{titlesec}
266 \LWR@loadaft{titletoc}
267 \LWR@loadaft{titling}
268 \LWR@loadaft{tocbibind}
269 \LWR@loadaft{tocloft}
270 \LWR@loadaft{transparent}
271 \LWR@loadaft{trivfloat}
272 \LWR@loadaft{typearea}
273 \LWR@loadaft{ulem}
274 \LWR@loadaft{upref}
275 \LWR@loadaft{variorref}
276 \LWR@loadaft{verse}
277 \LWR@loadaft{wallpaper}
278 \LWR@loadaft{wrapfig}
279 \LWR@loadaft{xcolor}
280 \LWR@loadaft{xfrac}
281 \LWR@loadaft{xltextra}
282 \LWR@loadaft{xmpincl}
283 \LWR@loadaft{xtab}
284 \LWR@loadaft{zwpagelayout}

```

21 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.

In the document preamble, create a `warpprint` environment, and place inside it any of the following packages which are required and which are labeled as “Print: OK to Load in a `warpprint` environment”. Those packages which are labeled as “Print: Pre-Loaded” need not be placed into the document preamble.

for HTML & PRINT: 285 `\begin{warppall}`

See: <http://tex.stackexchange.com/a/47579>.

Detects X_ET_EX and LuaX_ET_EX:

```

286 \RequirePackage{ifutex}
287 \newif\ifxetexorluatex

```

```

288 \ifXeTeX
289     \xetexorluatexttrue
290 \else
291     \ifLuaTeX
292         \xetexorluatexttrue
293     \else
294         \xetexorluatextfalse
295     \fi
296 \fi

297 \end{warpall}

for HTML output: 298 \begin{warpHTML}

299 \ifxetexorluatex
300 % ^~A      \usepackage[no-math]{fontspec}

```

The monospaced font is used for HTML tags, so turn off its TeX ligatures and common ligatures:

```

301 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
302 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
303 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
304 \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 L^ET_EX dash and quote ligatures, which may fail on older browsers but at least won't corrupt written words.

```

305 \RequirePackage{microtype}
306
307 \microtypesetup{
308     protrusion=false,
309     expansion=false,
310     tracking=false,
311     kerning=false,
312     spacing=false}
313
314 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

315 \fi

316 \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: 317 `\begin{warpHTML}`
318 `\RequirePackage[paperheight=190in,paperwidth=20in,%`
319 `left=2in,right=12in,%`
320 `top=1in,bottom=1in,%`
321 `]{geometry}`
322 `\@twosidefalse`
323 `\@mparswitchfalse`
324 `\end{warpHTML}`

for HTML & PRINT: 325 `\begin{warpall}`

Pkg `xparse`

TeX3 command argument parsing

326 `\RequirePackage{xparse}`

Pkg `afterpackage` Used to patch titling to add `\AddSubtitlePublished`.

327 `\RequirePackage{afterpackage}`

328 `\end{warpall}`

for HTML output: 329 `\begin{warpHTML}`

Pkg `expl3`

TeX3 programming

330 `\RequirePackage{expl3}`

Pkg `gettitlestring`

Used to emulate `\nameref`.

331 `\RequirePackage{gettitlestring}`

```
Pkg everyhook
everyhook is used to patch paragraph handling.

332 \RequirePackage{everyhook}
333 \end{warpHTML}

for HTML & PRINT: 334 \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

335 \RequirePackage{filecontents}
336
337 \@ifpackagelater{filecontents}{2011/10/09}%
338 {}
339 {
340 \newwrite\fcwrite
341 \let\LWR@origfilec@ntents\filec@ntents
342 \def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
343 }

344 \end{warpall}

for HTML output: 345 \begin{warpHTML}

Pkg xifthen
346 \RequirePackage{xifthen}

Pkg xstring
347 \RequirePackage{xstring}

Pkg xstring
348 \RequirePackage{verbatim}

Pkg makeidx
349 \RequirePackage{makeidx}
350 \makeindex
```

```
Pkg calc
351 \RequirePackage{calc}

Pkg refcount
352 \RequirePackage{refcount}

Pkg newfloat
353 \RequirePackage{newfloat}

Pkg caption
354 \RequirePackage{caption}

355 \end{warpHTML}

for HTML & PRINT: 356 \begin{warpall}

Pkg environ
Used to encapsulate math environments for re-use in HTML ALT text.

357 \RequirePackage{environ}

358 \end{warpall}

for HTML output: 359 \begin{warpHTML}

Pkg zref
Used for cross-references.

360 \RequirePackage{zref}

Pkg amsmath
Equation numbers are placed to the left for HTML.

newtxmath automatically loads amsmath, so the options leqno and fleqn are passed beforehand to be picked up both here and by newtxmath if it is used.

361 \PassOptionsToPackage{leqno}{amsmath}
362 \RequirePackage{amsmath}
```

Pkg xfrac

Patched for HTML use. See section 229.

363 \RequirePackage{xfrac}

Used to convert lengths for image width/height options.

364 \RequirePackage{printlen}

365 \end{warpHTML}

22 Loading packages

for HTML output: 366 \begin{warpHTML}

Remember the original \RequirePackage:

367 \LetLtxMacro{\LWR@origRequirePackage}{\RequirePackage}

\LWR@requirepackagenames Stores the list of required package names.

368 \newcommand*\LWR@requirepackagenames{}{}

\LWR@findword [*1: separator*] [*2: list*] [*3: index*] [*4: destination*]

Note that argument 4 is passed directly to \StrBetween.

369 \newcommand*\LWR@findword[3][,]{%
370 \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
371 }

\LWR@lookforpackagename [*index*] If this is a package name, re-direct it to the lwarp version by renaming it lwarp- followed by the original name.

372 \newcommand*\LWR@lookforpackagename[1]{%

Find the n'th package name from the list:

373 \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.

374 \StrSubstitute[100]{\LWR@strresult}{ }{}[\LWR@strresulttwo]{%

See if the package name was found:

```
375 \IfStrEq{\LWR@strresulttwo}{}
376 {}% no filename
377 {}% yes filename
```

If found, and if an l warp-equivalent name exists, use l warp-* instead.

```
378 \IfFileExists{l warp-\LWR@strresulttwo.sty}{}
379 {}% l warp-* file found
380     \StrSubstitute{%
381         {\LWR@requirepackagenames}%
382         {\LWR@strresult}%
383         {l warp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
384 }%
385 {}% no l warp-* file
386 {}% yes filename
387 }
```

\RequirePackage [⟨1: options⟩] {⟨2: package names⟩} [⟨3: version⟩]

For each of many package names in a comma-separated list, if an l warp version of a package exists, select it instead of the L^AT_EX version.

```
388 \RenewDocumentCommand{\RequirePackage}{o m o}{%
```

Redirect up to nine names:

```
389 \renewcommand*{\LWR@requirepackagenames}{#2}
390 \LWR@lookforpackagename{1}
391 \LWR@lookforpackagename{2}
392 \LWR@lookforpackagename{3}
393 \LWR@lookforpackagename{4}
394 \LWR@lookforpackagename{5}
395 \LWR@lookforpackagename{6}
396 \LWR@lookforpackagename{7}
397 \LWR@lookforpackagename{8}
398 \LWR@lookforpackagename{9}
```

\RequirePackage depending on the options and version:

```
399 \IfValueTF{#1}
400 {}% options given
401     \IfValueTF{#3}{% version given?
402         {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
403         {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
404 }
405 {}% no options given
```

```

406     \IfValueTF{#3}{% version given?
407     {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
408     {\LWR@origRequirePackage{\LWR@requirepackagenames}}
409 }
410 }
411 \LetLtxMacro{\usepackage}{\RequirePackage}

```

\LWR@ProvidesPackagePass {*pkgname*} [*version*]

Uses the original package, including options.

```

412 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
413 \PackageInfo{lwarp}{Using package '#1' and adding lwarp modifications, including options,}%
414 \IfValueTF{#2}{%
415 {\ProvidesPackage{lwarp-#1}[#2]}%
416 {\ProvidesPackage{lwarp-#1}}%
417 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}%
418 \ProcessOptions\relax
419 }
420 \IfValueTF{#2}{%
421 {\LWR@origRequirePackage{#1}[#2]}%
422 {\LWR@origRequirePackage{#1}}%
423 }

```

\LWR@ProvidesPackageDrop {*pkgname*} [*version*]

Ignores the original package and uses lwarp's version instead. Drops/discards all options.

```

424 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o} {
425 \PackageInfo{lwarp}{Replacing package '#1' with the lwarp version, discarding options,}%
426 \IfValueTF{#2}{%
427 {\ProvidesPackage{lwarp-#1}[#2]}%
428 {\ProvidesPackage{lwarp-#1}}%
429 \DeclareOption*{}%
430 \ProcessOptions\relax
431 }

432 \end{warpHTML}

```

23 File handles

Defines file handles for writes.

for HTML & PRINT: 433 \begin{warpall}

\LWR@quickfile For quick temporary use only. This is reused in several places.

```
434 \newwrite\LWR@quickfile%
435 \end{warpall}
```

for HTML output: 436 \begin{warpHTML}

\LWR@lateximagesfile For lateximages.txt.

```
437 \newwrite\LWR@lateximagesfile
438 \end{warpHTML}
```

24 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 L^AT_EX.

for HTML output: 439 \begin{warpHTML}

```
\include {\langle filename\rangle}
```

\@include {\langle filename\rangle} Modified to load _html_inc files.

```
440 \def\@include#1 {%
441 \immediate\openout\LWR@quickfile #1_html_inc.tex% new
442 \immediate\write\LWR@quickfile{\string\input{\#1.tex}}% new
443 \immediate\closeout\LWR@quickfile% new
444 \LWR@origclearpage% \changed
445 \if@files
446   \immediate\write@mainaux{\string\@input{\#1_html_inc.aux}}% changed
447 \fi
448 \tempswattrue
449 \if@partsw
450   \tempswafalse
```

```

451     \edef\reserved@b{\#1}%
452     \for\reserved@a:=\partlist\do
453     {\ifx\reserved@a\reserved@b\@tempswtrue\fi}%
454 \fi
455 \if@tempswa
456   \let\auxout\partaux
457   \if@files w
458     \immediate\openout\partaux #1_html_inc.aux % changed
459     \immediate\write\partaux{\relax}%
460   \fi
461   \input{\#1_html_inc.tex} % changed
462   \LWR@origclearpage% changed
463   \writeckpt{\#1}%
464   \if@files w
465     \immediate\closeout\partaux
466   \fi
467 \else
468   \deadcycles{z@}
469   \nameuse{cp{\#1}}%
470 \fi
471 \let\auxout\mainaux%
472 }

473 \end{warpHTML}

```

25 Copying a file

for HTML output: 474 \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.

```

475 \newwrite\LWR@copyoutfile % open the file to write to
476 \newread\LWR@copyinfile % open the file to read from
477
478 \newcommand*{\LWR@copyfile}[2]{%
479   \immediate\openout\LWR@copyoutfile=\#2
480   \openin\LWR@copyinfile=\#1
481   \begingroup\endlinechar=-1
482   \makeatletter
483   \loop\unless\ifeof\LWR@copyinfile
484     \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
485     \LWR@fileline\par % print the content into the pdf
486   % print the content:

```

```

487  \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
488 \repeat
489 \closeout\LWR@copyoutfile
490 \endgroup
491 }

492 \end{warpHTML}

```

26 Debugging messages

```
493 \begin{warpall}
```

Bool LWR@tracinglwarp True if tracing is turned on.

```
494 \newbool{LWR@tracinglwarp}
```

\tracinglwarp Turns on the debug tracing messages.

```
495 \newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}
```

\LWR@traceinfo {\text} If tracing is turned on, writes the text to the .log file.

```

496 \newcommand{\LWR@traceinfo}[1]{%
497 \ifbool{LWR@tracinglwarp}{%
498 {%
499   \typeout{*** lwarp: #1}%
500   % \PackageInfo{lwarp}{#1 : }%
501 }%
502 {}%
503 }

```

Bool HTMLDebugComments Default false. Add comments in HTML about closing <div>s, sections, etc.

```

504 \newbool{HTMLDebugComments}
505 \boolfalse{HTMLDebugComments}

```

If \tracinglwarp, show where preamble hooks occur:

```

506 \AfterEndPreamble{
507 \LWR@traceinfo{AfterEndPreamble}
508 }
509
510 \AtBeginDocument{
511 \LWR@traceinfo{AtBeginDocument}
512 }

```

27 HTML-conversion output modifications

These booleans modify the HTML output in various ways to improve conversion to EPUB or word processor imports.

`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.

```
513 \newbool{FormatEPUB}
514 \boolfalse{FormatEPUB}
```

`Bool FormatWordProcessor` 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.

```
515 \newbool{FormatWordProcessor}
516 \boolfalse{FormatWordProcessor}
```

`Bool HTMLMarkFloats` Default true. Adds

```
==== table begin
...
==== end
```

or

```
==== figure begin
...
==== end
```

around floats while formatting for word processors. This helps identify boundaries of floats to be manually converted to word-processor frames.¹²

```
517 \newbool{HTMLMarkFloats}
518 \booltrue{HTMLMarkFloats}

519 \end{warpall}
```

28 Remembering original formatting macros

for HTML output: 520 \begin{warpHTML}

¹²Perhaps some day word processors will have HTML import options for identifying <div> classes for figures and tables, along with the associated captions.

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.

```
521 \LetLtxMacro{\LWR@origtextrm}{\textrm}
522 \LetLtxMacro{\LWR@origtextsfs}{\textsf}
523 \LetLtxMacro{\LWR@origtexttt}{\texttt}
524 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
525 \LetLtxMacro{\LWR@origtextbf}{\textbf}
526 \LetLtxMacro{\LWR@origtextmd}{\textmd}
527 \LetLtxMacro{\LWR@origtextit}{\textit}
528 \LetLtxMacro{\LWR@origtextsfs}{\textsf}
529 \LetLtxMacro{\LWR@origtextsc}{\textsc}
530 \LetLtxMacro{\LWR@origtextup}{\textup}
531 \LetLtxMacro{\LWR@origemph}{\emph}
532
533 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
534 \LetLtxMacro{\LWR@origsffamily}{\sffamily}
535 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
536 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
537 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
538 \LetLtxMacro{\LWR@origupshape}{\upshape}
539 \LetLtxMacro{\LWR@origslshape}{\slshape}
540 \LetLtxMacro{\LWR@origscshape}{\scshape}
541 \LetLtxMacro{\LWR@origitshape}{\itshape}
542 \LetLtxMacro{\LWR@origem}{\em}
543 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
544
545 \let\LWR@origraggedright\raggedright
546 \let\LWR@origonecolumn\onecolumn
547
548 \let\LWR@origsp\sp
549 \let\LWR@origsb\sb
550 \let\LWR@origtextsuperscript\textsuperscript
551 \let\LWR@origtextsubscript\textsubscript
552
553 \let\LWR@origscriptsize\scriptsize
554
555 \let\LWR@orignewpage\newpage
556
557 \let\LWR@origpagestyle\pagestyle
558 \let\LWR@origthispagestyle>thispagestyle
559 \let\LWR@origpagenumbering\pagenumbering
560
561 \LetLtxMacro{\LWR@origminipage}{\minipage}
562 \let\LWR@origendminipage\endminipage
563 \LetLtxMacro{\LWR@origparbox}{\parbox}
564
565 \let\LWR@orignewline\newline
```

```

566
567 \LetLtxMacro{\LWR@origitem}{\item}
568
569 \LetLtxMacro{\LWR@origitemize}{\itemize}
570 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
571 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
572 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
573 \LetLtxMacro{\LWR@origdescription}{\description}
574 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
575
576
577 \let\LWR@origpar\par
578
579
580 \LetLtxMacro{\LWR@origfootnote}{\footnote}
581 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
582
583 \let\LWR@origclearpage\clearpage
584 \let\clearpage\relax
585 \let\cleardoublepage\relax

586 \end{warpHTML}

```

29 Accents

Native \TeX accents such as \^n will work, but many more kinds of accents are available when using Unicode-aware $\text{X}\text{\TeX}$ and $\text{Lua}\text{\TeX}$.

for HTML output: 587 \begin{warpHTML}

Without \AtBeginDocument , \t was being re-defined somewhere.

```
588 \AtBeginDocument{
```

The following are restored for print when inside a \teximage .

For Unicode engines, only \t needs to be redefined:

```
589 \LetLtxMacro{\LWR@origt}{\t}
```

For $\text{pdf}\text{\TeX}$, additional work is required:

```

590 \ifPDFTeX
591 \LetLtxMacro{\LWR@origequalaccent}{\=}
592 \LetLtxMacro{\LWR@origdotaccent}{\.\.}
593 \LetLtxMacro{\LWR@origu}{\u}
```

```

594 \LetLtxMacro{\LWR@origv}{\v}
595 \LetLtxMacro{\LWR@origc}{\c}
596 \LetLtxMacro{\LWR@origd}{\d}
597 \LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For pdf \TeX , Unicode diacritical marks are used:

```

598 \renewcommand*{\=}[1]{\#1\HTMLunicode{0305}}
599 \renewcommand*{\.}[1]{\#1\HTMLunicode{0307}}
600 \renewcommand*{\u}[1]{\#1\HTMLunicode{0306}}
601 \renewcommand*{\v}[1]{\#1\HTMLunicode{030C}}
602 \renewcommand*{\c}[1]{\#1\HTMLunicode{0327}}
603 \renewcommand*{\d}[1]{\#1\HTMLunicode{0323}}
604 \renewcommand*{\b}[1]{\#1\HTMLunicode{0331}}
605 \fi
```

For all engines, a Unicode diacritical tie is used:

```

606 \def\LWR@t#1#2{\#1\HTMLunicode{0361}#2}
607 \renewcommand*{\t}[1]{\LWR@t#1}
```

`\LWR@restoreorigaccents` Called from `\restoreoriginalformatting` when a `lateximage` is begun.

```

608 \ifPDFTeX
609 \newcommand*{\LWR@restoreorigaccents}{%
610 \LetLtxMacro{\=}{\LWR@origequalaccent}%
611 \LetLtxMacro{\.}{\LWR@origdotaccent}%
612 \LetLtxMacro{\u}{\LWR@origu}%
613 \LetLtxMacro{\v}{\LWR@origu}%
614 \LetLtxMacro{\t}{\LWR@origt}%
615 \LetLtxMacro{\c}{\LWR@origc}%
616 \LetLtxMacro{\d}{\LWR@origd}%
617 \LetLtxMacro{\b}{\LWR@origb}%
618 }
619 \else% XeLaTeX, LuaLaTeX:
620 \newcommand*{\LWR@restoreorigaccents}{%
621 \LetLtxMacro{\t}{\LWR@origt}%
622 }
623 \fi
624 }% AtBeginDocument
```

```
625 \end{warpHTML}
```

30 Configuration Files

```
626 \begin{warpprint}
627 \typeout{l warp: generating configuration files}
628 \end{warpprint}
```

30.1 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.

```
629 \begin{warpprint}
630 \immediate\openout\LWR@quickfile=\jobname_html.tex
631 \immediate\write\LWR@quickfile{%
632 \detokenize{\PassOptionsToPackage}{%
633 {warpHTML,BaseJobname=\jobname}\{l warp\}}%
634 }
635 \immediate\write\LWR@quickfile{%
636 \detokenize{\input}\string{\jobname.tex}\string }%
637 }
638 \immediate\closeout\LWR@quickfile
639 \end{warpprint}
```

30.2 lwarpmk.conf

File `lwarpmk.conf` `lwarpmk.conf` is automatically (re-)created by the `l warp` 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`.

An example `lwarpmk.conf`:

```
opsystem = "Unix" -- or "Windows"
latexname = "pdflatex" -- or "lualatex" or "xelatex"
sourcename = "projectname" -- your .tex source
homehtmlfilename = "index" -- or "projectname"
htmlfilename = "" -- or "projectname" if numbered HTML files
```

for PRINT output:

```
640 \begin{warpprint}
641 \ifcsdef{\LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
642 \immediate\openout\LWR@quickfile=lwarpmk.conf
643 \ifbool{usingOSWindows}{
```

```

644 \immediate\write\LWR@quickfile{opsystem = "Windows"}
645 }{
646 \immediate\write\LWR@quickfile{opsystem = "Unix"}
647 }
648 \ifPDFTeX
649 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
650 \fi
651 \ifXeTeX
652 \immediate\write\LWR@quickfile{latexname = "xelatex"}
653 \fi
654 \ifLuaTeX
655 \immediate\write\LWR@quickfile{latexname = "lualatex"}
656 \fi
657 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
658 \immediate\write\LWR@quickfile{%
659 homehtmlfilename = "\HomeHTMLFilename"%
660 }
661 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
662 \immediate\write\LWR@quickfile{latexmk = "\ifbool{\LWR@latexmk}{true}{false}"}
663 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
664 \immediate\write\LWR@quickfile{xdfyfile = "\LWR@xdfyFilename"}
665 \immediate\closeout\LWR@quickfile
666 \end{warpprint}

```

30.3 project.lwarpmkconf

File project.lwarpmkconf A project-specific configuration file for lwarpmk.

```

667 \begin{warpprint}
668 \ifcsdef{\LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
669 \immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
670 \ifbool{usingOSWindows}{%
671 \immediate\write{\LWR@quickfile{opsystem = "Windows"}}
672 }{%
673 \immediate\write{\LWR@quickfile{opsystem = "Unix"}}
674 }
675 \ifPDFTeX
676 \immediate\write{\LWR@quickfile{latexname = "pdflatex"}}
677 \fi
678 \ifXeTeX
679 \immediate\write{\LWR@quickfile{latexname = "xelatex"}}
680 \fi
681 \ifLuaTeX
682 \immediate\write{\LWR@quickfile{latexname = "lualatex"}}
683 \fi
684 \immediate\write{\LWR@quickfile{sourcename = "\jobname"}}
685 \immediate\write{\LWR@quickfile{%

```

```
686 homehtmlfilename = "\HomeHTMLFilename"%
687 }
688 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
689 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexm}{true}{false}"}
690 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
691 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
692 \immediate\closeout\LWR@quickfile
693 \end{warpprint}
```

30.4 lwarf.css

File `lwarf.css` This is the base css layer used by lwarf.

This must be present both when compiling the project and also when distributing the HTML files.

```
694 \begin{warpprint}
695 \begin{filecontents*}{lwarf.css}
696 /*
697   CSS stylesheet for the LaTeX lwarf package
698   Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
699 */
700
701
702 /* a fix for older browsers: */
703 header, section, footer, aside, nav, main,
704     article, figure { display: block; }
705
706
707 A:link {color:#000080 ; text-decoration: none ; }
708 A:visited {color:#800000 ; }
709 A:hover {color:#000080 ; text-decoration: underline ;}
710 A:active {color:#800000 ; }
711
712 a.tocpart {display: inline-block ; margin-left: 0em ;
713     font-weight: bold ;}
714 a.tocchapter {display: inline-block ; margin-left: 0em ;
715     font-weight: bold ;}
716 a.tocsection {display: inline-block ; margin-left: 1em ;
717     text-indent: -.5em ; font-weight: bold ;}
718 a.tocsubsection {display: inline-block ; margin-left: 2em ;
719     text-indent: -.5em ;}
720 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
721     text-indent: -.5em ;}
722 a.tocparagraph {display: inline-block ; margin-left: 4em ;
723     text-indent: -.5em ;}
724 a.toc subparagraph {display: inline-block ; margin-left: 5em ;}
```

```
725     text-indent: -.5em ; }
726 a.tocfigure {margin-left: 0em}
727 a.tocsubfigure {margin-left: 2em}
728 a.toctable {margin-left: 0em}
729 a.tocsubtable {margin-left: 2em}
730 a.toctheorem {margin-left: 0em}
731 a.toclstlisting {margin-left: 0em}
732
733
734 body {
735     font-family: "DejaVu Serif", "Bitstream Vera Serif",
736             "Lucida Bright", Georgia, serif;
737     background: #FAF7F4 ;
738     color: black ;
739     margin: 0em ;
740     padding: 0em ;
741     font-size: 100% ;
742     line-height: 1.2 ;
743 }
744
745 p {margin: 1.5ex 0em 1.5ex 0em ;}
746
747 /* Holds a section number to add space between it and the name */
748 span.sectionnumber { margin-right: 0em }
749
750 /* Inserted in front of index lines */
751 span.indexitem {margin-left: 0em}
752 span.indexsubitem {margin-left: 2em}
753 span.indexsubsubitem {margin-left: 4em}
754
755 div.hidden, span.hidden { display: none ; }
756
757 kbd {
758     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
759             "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
760             "Courier New", monospace;
761     font-size: 100% ;
762 }
763
764 pre { padding: 3pt ; }
765
766 span.strong { font-weight: bold; }
767
768 span.textmd { font-weight: normal; }
769
770 span.textsc { font-variant: small-caps; }
771
772 span.textsl { font-style: oblique; }
773
774 span.textup { font-variant: normal; }
```

```
775
776 span.textrm {
777     font-family: "DejaVu Serif", "Bitstream Vera Serif",
778     "Lucida Bright", Georgia, serif;
779 }
780
781 span.textsf {
782     font-family: "DejaVu Sans", "Bitstream Vera Sans",
783     Geneva, Verdana, sans-serif ;
784 }
785
786 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
787
788
789
790 /* For realscripts */
791 .supsubscript {
792     display: inline-block;
793     text-align:left ;
794 }
795
796 .supsubscript sup,
797 .supsubscript sub {
798     position: relative;
799     display: block;
800     font-size: .5em;
801     line-height: 1;
802 }
803
804 .supsubscript sup {
805     top: .5em;
806 }
807
808 .supsubscript sub {
809     top: .5em;
810 }
811
812
813
814
815 span.attribution {
816     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
817 }
818
819 span.citetitle {
820     margin-left: 1em ; font-size: 80% ; font-style: oblique;
821 }
822
823 span.poemtitle {
824     font-size: 120% ; font-weight: bold;
```

```
825 }
826
827blockquote {
828     margin-left: 0px ;
829     margin-right: 0px ;
830 }
831
832blockquote p {
833     line-height: 1.5;
834     text-align: left ;
835     font-size: .85em ;
836     margin-left: 3em ;
837     margin-right: 3em ;
838 }
839
840blockquotation {
841     margin-left: 0px ;
842     margin-right: 0px ;
843 }
844
845blockquotation p {
846     line-height: 1.5;
847     text-align: left ;
848     font-size: .85em ;
849     margin-left: 3em ;
850     margin-right: 3em ;
851 }
852
853div.epigraph {
854     line-height: 1.2;
855     text-align: left ;
856     padding: 3ex 1em 0ex 1em ;
857 /*      margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
858     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
859 /*      margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
860     font-size: .85em ;
861     max-width: 27em ;
862 }
863
864
865
866div.epigraphsource{
867     text-align:right ;
868     margin-left:auto ;
869 /*      max-width: 50% ; */
870     border-top: 1px solid #A0A0A0 ;
871     padding-bottom: 3ex ;
872     line-height: 1.2;
873 }
874
```

```
875 div.epigraph p { padding: .5ex ; margin: 0ex ; }
876 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ; }
877
878
879 /* lettrine package: */
880 span.lettrine { font-size: 3ex ; float: left ; }
881 span.lettrinetext { font-variant: small-caps ; }
882
883 /* ulem and soul packages: */
884 span.uline {
885     text-decoration: underline ;
886     text-decoration-skip ;
887 }
888
889 span.uline {
890     text-decoration: underline ;
891     text-decoration-skip ;
892     text-decoration-style: double ;
893 }
894
895 span.uwave {
896     text-decoration: underline ;
897     text-decoration-skip ;
898     text-decoration-style: wavy ;
899 }
900
901 span.sout {
902     text-decoration: line-through ;
903 }
904
905 span.xout {
906     text-decoration: line-through ;
907 }
908
909 span.dashuline {
910     text-decoration: underline ;
911     text-decoration-skip ;
912     text-decoration-style: dashed ;
913 }
914
915 span.dotuline {
916     text-decoration: underline ;
917     text-decoration-skip ;
918     text-decoration-style: dotted ;
919 }
920
921 span.letterspacing { letter-spacing: .2ex ; }
922
923 span.capsspacing {
924     font-variant: small-caps ;
```

```
925     letter-spacing: .1ex ;
926 }
927
928 span.highlight { background: #F8E800 ; }
929
930
931
932
933 html body {
934   margin: 0 ;
935   line-height: 1.2;
936 }
937
938
939 body div {
940   margin: 0ex;
941 }
942
943
944 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
945 {
946   font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
947   "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
948   "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
949   "Times New Roman", serif;
950   font-style: normal ;
951   font-weight: bold ;
952   text-align: left ;
953 }
954
955 h1 { /* title of the entire website, used on each page */
956   text-align: center ;
957   font-size: 2.5em ;
958   padding: .4ex 0em 0ex 0em ;
959 }
960 h2 { font-size: 2.25em }
961 h3 { font-size: 2em }
962 h4 { font-size: 1.75em }
963 h5 { font-size: 1.5em }
964 h6 { font-size: 1.25em }
965 span.paragraph {font-size: 1em ; font-variant: normal ;
966   margin-right: 1em ; }
967 span.subparagraph {font-size: 1em ; font-variant: normal ;
968   margin-right: 1em ; }
969
970
971
972 /* Title of the file */
973 h1 {
974   margin: 0ex 0em 0ex 0em ;
```

```
975   line-height: 1.3;
976   text-align: center ;
977 }
978
979 /* Part */
980 h2 {
981   margin: 1ex 0em 1ex 0em ;
982   line-height: 1.3;
983   text-align: center ;
984 }
985
986 /* Chapter */
987 h3 {
988   margin: 3ex 0em 1ex 0em ;
989   line-height: 1.3;
990 }
991
992 /* Section */
993 h4 {
994   margin: 3ex 0em 1ex 0em ;
995   line-height: 1.3;
996 }
997
998 /* Sub-Section */
999 h5 {
1000   margin: 3ex 0em 1ex 0em ;
1001   line-height: 1.3;
1002 }
1003
1004 /* Sub-Sub-Section */
1005 h6 {
1006   margin: 3ex 0em 1ex 0em ;
1007   line-height: 1.3;
1008 }
1009
1010
1011 div.titlepage {
1012   text-align: center ;
1013 }
1014
1015 .footnotes {
1016   font-size: .85em ;
1017   margin: 3ex 1em 0ex 1em ;
1018   padding-bottom: 1ex ;
1019   border-top: 1px solid silver ;
1020 }
1021
1022 .marginpar {
1023   max-width:50%;
1024   float:right;
```

```
1025     text-align:left;
1026     margin: 1ex 0.5em 1ex 1em ;
1027     padding: 1ex 0.5em 1ex 0.5em ;
1028     font-size: 85% ;
1029     border-top: 1px solid silver ;
1030     border-bottom: 1px solid silver ;
1031     overflow-x: auto;
1032 }
1033
1034 .marginpar br { margin-bottom: 2ex ; }
1035
1036 div.marginblock {
1037     max-width:50%;
1038     float:right;
1039     text-align:left;
1040     margin: 1ex 0.5em 1ex 1em ;
1041     padding: 1ex 0.5em 1ex 0.5em ;
1042     overflow-x: auto;
1043 }
1044
1045 div.marginblock div.minipage {
1046     display: block ;
1047     margin: 0pt auto 0pt auto ;
1048 }
1049
1050 div.marginblock div.minipage p { font-size: 85%}
1051
1052 div.marginblock br { margin-bottom: 2ex ; }
1053
1054
1055 section.textbody div.footnotes{
1056     margin: 3ex 0em 0ex 0em ;
1057     border-bottom: 2px solid silver ;
1058 }
1059
1060 .footnoteheader {
1061     border-top: 2px solid silver ;
1062     margin-top: 3ex ;
1063     padding-top: 1ex ;
1064     font-weight: bold ;
1065 }
1066
1067 .mpfootnotes {
1068     text-align: left ;
1069     font-size: .85em ;
1070     margin-left: 1em ;
1071     border-top: 1px solid silver ;
1072 }
1073
1074 /* Remove footnote top border in the title page. */
```

```
1075 div.titlepage div.mpfootnotes {  
1076     border-top: none ;  
1077 }  
1078  
1079  
1080  
1081 ol {  
1082     margin: 1ex 1em 1ex 0em;  
1083     line-height: 1.2;  
1084 }  
1085  
1086 ul, body dir, body menu {  
1087     margin: 1ex 1em 1ex 0em;  
1088     line-height: 1.2;  
1089 }  
1090  
1091 li { margin: 0ex 0em 1ex 0em; }  
1092  
1093 html {  
1094     margin: 0;  
1095     padding: 0;  
1096 }  
1097  
1098 .programlisting {  
1099     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",  
1100         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",  
1101         "Courier New", monospace;  
1102     margin: 1ex 0ex 1ex 0ex ;  
1103     padding: .5ex 0pt .5ex 0pt ;  
1104     overflow-x: auto;  
1105 }  
1106  
1107 section.textbody>pre.programlisting {  
1108 border-top: 1px solid silver ;  
1109 border-bottom: 1px solid silver ;  
1110 }  
1111  
1112  
1113 .inlineprogramlisting {  
1114     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",  
1115         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",  
1116         "Courier New", monospace;  
1117     overflow-x: auto;  
1118 }  
1119  
1120 span.listinglabel {  
1121     display: inline-block ;  
1122     font-size: 70% ;  
1123     width: 4em ;  
1124     text-align: right ;
```

```
1125     margin-right: 2em ;
1126 }
1127
1128 div.abstract {
1129   margin: 2em 5% 2em 5% ;
1130   padding: 1ex 1em 1ex 1em ;
1131 /* font-weight: bold ; */
1132   font-size: 90% ;
1133   text-align: left ;
1134 }
1135
1136 div.abstract dl {line-height:1.5;}
1137 div.abstract dt {color:#304070;}
1138
1139 div.abstracttitle{
1140   font-family: "URW Classico", Optima, "Linux Biolinum O",
1141     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1142     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1143   font-weight:bold;
1144   font-size:1.25em;
1145   text-align: center ;
1146 }
1147
1148 span.abstractrunintitle{
1149   font-family: "URW Classico", Optima, "Linux Biolinum O",
1150     "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1151     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1152   font-weight:bold;
1153 }
1154
1155
1156 .verbatim {
1157   overflow-x: auto ;
1158 }
1159
1160 .alltt {
1161   overflow-x: auto ;
1162 }
1163
1164
1165 .bverbatim {
1166   margin: 1ex Opt 1ex Opt ;
1167   padding: .5ex Opt .5ex Opt ;
1168   overflow-x: auto ;
1169 }
1170
1171 .lverbatim {
1172   margin: 1ex Opt 1ex Opt ;
1173   padding: .5ex Opt .5ex Opt ;
1174   overflow-x: auto ;
```

```
1175 }
1176
1177 .fancyvrb {
1178     font-size:.85em ;
1179     margin: 3ex 0pt 3ex 0pt
1180 }
1181
1182 .fancyvrblabel {
1183     font-weight:bold;
1184     text-align: center ;
1185 }
1186
1187
1188 .verse {
1189     font-family: "Linux Libertine Mono O", "Lucida Console",
1190             "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1191             "Liberation Mono", "FreeMono", "Andale Mono",
1192             "Nimbus Mono L", "Courier New", monospace;
1193     margin-left: 1em ;
1194 }
1195
1196
1197 div.singlespace { line-height: 1.2 ; }
1198 div.onehalfspace { line-height: 1.5 ; }
1199 div.doublespace { line-height: 2 ; }
1200
1201
1202
1203
1204
1205 /* Minipage environments, vertically aligned to top, center, bottom: */
1206 .minipage, .fminipage, .fcolorminipage {
1207     /* display: inline-block ; */
1208     /* Mini pages which follow each other will be tiled. */
1209     margin: .25em .25em .25em .25em;
1210     padding: .25em .25em .25em .25em;
1211     display: inline-flex;
1212     flex-direction: column ;
1213     overflow: auto;
1214 }
1215
1216 /* Paragraphs in the flexbox did not collapse their margins. */
1217 /* Have not yet researched this. */
1218 .minipage p {margin: .75ex 0em .75ex 0em ;}
1219
1220 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1221 .fcolorbox .minipage, .fcolorboxBlock .minipage
1222     {border: none ; background: none;}
1223
1224 .fbox, .fboxBlock { border: 1px solid black ; }
```

```
1225
1226 .fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1227 .fminipage, .fcolorminipage
1228     {display: inline-block}
1229
1230 .shadowbox, .shabox {
1231     border: 1px solid black;
1232     box-shadow: 3px 3px 3px #808080 ;
1233     border-radius: 0px ;
1234     padding: .4ex .3em .4ex .3em ;
1235     margin: 0pt .3ex 0pt .3ex ;
1236     display: inline-block ;
1237 }
1238
1239 .doublebox {
1240     border: 3px double black;
1241     border-radius: 0px ;
1242     padding: .4ex .3em .4ex .3em ;
1243     margin: 0pt .3ex 0pt .3ex ;
1244     display: inline-block ;
1245 }
1246
1247 .ovalbox, .Ovalbox {
1248     border: 1px solid black;
1249     border-radius: 1ex ;
1250     padding: .4ex .3em .4ex .3em ;
1251     margin: 0pt .3ex 0pt .3ex ;
1252     display: inline-block ;
1253 }
1254
1255 .Ovalbox { border-width: 2px ; }
1256
1257 .framebox {
1258     border: 1px solid black;
1259     border-radius: 0px ;
1260     padding: .3ex .2em 0ex .2em ;
1261     margin: 0pt .1ex 0pt .1ex ;
1262     display: inline-block ;
1263 }
1264
1265
1266 .mdframed {
1267 /*     padding: 0ex ; */
1268 /*     border: 1px solid blafck; */
1269 /*     border-radius: 0px ; */
1270     padding: 0ex ;
1271     margin: 3ex 5% 3ex 5% ;
1272 /*     display: inline-block ; */
1273 }
1274
```

```
1275 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1276
1277 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1278
1279 .mdframedtitle {
1280     padding: .5em ;
1281     display: block ;
1282     font-size: 130%
1283 }
1284
1285 .mdframedsubtitle {
1286     padding: 0ex .5em 0ex .5em ;
1287     display: block ;
1288     font-size: 115% ;
1289 }
1290
1291 .mdframedsubsubtitle {
1292     padding: 0ex .5em 0ex .5em ;
1293     display: block ;
1294 }
1295
1296 .mdtheorem {
1297     padding: 0ex .5em 0ex .5em ;
1298     margin: 3ex 5% 3ex 5% ;
1299 /*   display: inline-block ; */
1300 }
1301
1302
1303 /* framed package */
1304 .framed, pre.boxedverbatim, fcolorbox {
1305     margin: 3ex 0em 3ex 0em ;
1306     border: 1px solid black;
1307     border-radius: 0px ;
1308     padding: .3ex 1em 0ex 1em ;
1309     display: block ;
1310 }
1311
1312 .snugframed {
1313     margin: 3ex 0em 3ex 0em ;
1314     border: 1px solid black;
1315     border-radius: 0px ;
1316     display: block ;
1317 }
1318
1319 .framedleftbar {
1320     margin: 3ex 0em 3ex 0em ;
1321     border-left: 3pt solid black;
1322     border-radius: 0px ;
1323     padding: .3ex .2em .3ex 1em ;
1324     display: block ;
```

```
1325 }
1326
1327 .framedtitle {
1328     margin: 0em ;
1329     padding: 0em ;
1330     font-size: 130%
1331 }
1332
1333 .framedtitle p { padding: .3em }
1334
1335
1336
1337 dl {
1338     margin: 1ex 2em 1ex 0em;
1339     line-height: 1.3;
1340 }
1341
1342 dl dt {
1343     margin-top: 1ex;
1344     margin-left: 1em ;
1345     font-weight: bold;
1346 }
1347
1348 dl dd p { margin-top: 0em; }
1349
1350
1351 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1352     font-family: "URW Classico", Optima, "Linux Biolinum O",
1353             "DejaVu Sans", "Bitstream Vera Sans",
1354             Geneva, Verdana, sans-serif ;
1355     margin-bottom: 4ex ;
1356 }
1357
1358 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1359     line-height: 1.2 ;
1360     margin-top:.5ex ;
1361     margin-bottom:.5ex;
1362     font-size: .9em ;
1363 }
1364
1365
1366
1367 img, img.hyperimage, img.borderimage {
1368     max-width: 600px;
1369     border: 1px solid silver;
1370     box-shadow: 3px 3px 3px #808080 ;
1371     padding: .5% ;
1372     margin: .5% ;
1373     background: none ;
1374 }
```

```
1375
1376 img.inlineimage{
1377     padding: 0px ;
1378     box-shadow: none ;
1379     border: none ;
1380     background: none ;
1381     margin: 0px ;
1382     display: inline-block ;
1383     border-radius: 0px ;
1384 }
1385
1386 img.logoimage{
1387     max-width: 300px ;
1388     box-shadow: 3px 3px 3px #808080 ;
1389     border: 1px solid black ;
1390     background:none ;
1391     padding:0 ;
1392     margin:.5ex ;
1393     border-radius: 10px ;
1394 }
1395
1396
1397 .section {
1398 /*
1399     To have each section float relative to each other:
1400 */
1401 /*
1402     display: block ;
1403     float: left ;
1404     position: relative ;
1405     background: white ;
1406     border: 1px solid silver ;
1407     padding: .5em ;
1408 */
1409     margin: 0ex .5em 0ex .5em ;
1410     padding: 0 ;
1411 }
1412
1413
1414 figure {
1415     margin: 3ex auto 3ex auto ;
1416     padding: 1ex 1em 1ex 1em ;
1417     overflow-x: auto ;
1418 }
1419
1420
1421 /* To automatically center images in figures: */
1422 /*
1423 figure img.inlineimage {
1424     margin: 0ex auto 0ex auto ;
```

```
1425     display: block ;
1426 }
1427 */
1428
1429 /* To automatically center minipages in figures: */
1430 /*
1431 figure div.minipage, figure div.minipage div.minipage {
1432     margin: 1ex auto 1ex auto ;
1433     display: block ;
1434 }
1435 */
1436
1437 figure div.minipage p { font-size: 85% ; }
1438
1439 figure.subfigure, figure.subtable {
1440     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1441 }
1442
1443 figcaption .minipage { margin:0 ; padding: 0 }
1444
1445 div.minipage figure { border: none ; box-shadow: none ; }
1446
1447 div.floatrow { text-align: center; }
1448
1449 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1450
1451 div.floatfoot { font-size: .85em ;
1452     border-top: 1px solid silver ; line-height: 1.2 ; }
1453
1454 figcaption , .lstlistingtitle {
1455     font-size: .85em ;
1456     text-align: center ;
1457     font-weight: bold ;
1458     margin-top: 1ex ;
1459     margin-bottom: 1ex ;
1460 }
1461
1462 figure.subfigure figcaption, figure.subtable figcaption {
1463     border-bottom: none ; background: none ;
1464 }
1465
1466 div.nonfloatcaption {
1467     margin: 1ex auto 1ex auto ;
1468     font-size: .85em ;
1469     text-align: center ;
1470     font-weight: bold ;
1471 }
1472
1473 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1474 figure div.floatrow div.minipage figcaption {
```

```
1475     border: none ;
1476     background: none ;
1477 }
1478
1479
1480 table {
1481     margin: 1ex auto 1ex auto ;
1482     border-collapse: collapse ;
1483     border-spacing: 0px ;
1484     line-height: 1.3 ;
1485 }
1486
1487 tr.hline {border-top: 1px solid silver ; margin-top: 0ex ;
1488     margin-bottom: 0ex ; } /* for \hline */
1489
1490 tr.tbrule {border-top: 1px solid black ; margin-top: 0ex ;
1491     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1492
1493 td {padding: 1ex .5em 1ex .5em ;}
1494
1495 table td.tdl { text-align: left ; vertical-align: middle ; }
1496 table td.tdc { text-align: center ; vertical-align: middle ; }
1497 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1498 table td.tdbang { text-align: center ; vertical-align: middle ; }
1499 table td.tdr { text-align: right ; vertical-align: middle ; }
1500 table td.tdp { text-align: left ; vertical-align: bottom ; }
1501 table td.tdm { text-align: left ; vertical-align: middle ; }
1502 table td.tdb { text-align: left ; vertical-align: top ; }
1503 table td.tdP { text-align: center ; vertical-align: bottom ; }
1504 table td.tdM { text-align: center ; vertical-align: middle ; }
1505 table td.tdB { text-align: center ; vertical-align: top ; }
1506 table td.tdlrule { text-align: left ; border-top: 1px solid silver ;
1507     vertical-align: middle ; } /* for cmidrule */
1508 table td.tdcrule { text-align: center ; border-top: 1px solid silver ;
1509     vertical-align: middle ; }
1510 table td.tdatrule { text-align: center ; border-top: 1px solid silver ;
1511     vertical-align: middle ; padding: 0px ; margin: 0px ; }
1512 table td.tdbangrule { text-align: center ; border-top: 1px solid silver ;
1513     vertical-align: middle ; } /* for cmidrule */
1514 table td.tdrrule { text-align: right ; border-top: 1px solid silver ;
1515     vertical-align: middle ; }
1516 table td.tdprule { text-align: left ; border-top: 1px solid silver ;
1517     vertical-align: bottom ; }
1518 table td.tdmrule { text-align: left ; border-top: 1px solid silver ;
1519     vertical-align: middle ; }
1520 table td.tdbrule { text-align: left ; border-top: 1px solid silver ;
1521     vertical-align: top ; }
1522 table td.tdPrule { text-align: center ; border-top: 1px solid silver ;
1523     vertical-align: bottom ; }
1524 table td.tdMrule { text-align: center ; border-top: 1px solid silver ;
```

```
1525     vertical-align: middle ; }
1526 table td.tdBrule { text-align: center ; border-top: 1px solid silver ;
1527     vertical-align: top ; }
1528
1529 /* Margins of paragraphs inside table cells: */
1530 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1531     padding-bottom: 1ex ; margin: 0ex ; }
1532 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1533     padding-bottom: 1ex ; margin: 0ex ; }
1534 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1535     padding-bottom: 1ex ; margin: 0ex ; }
1536
1537 td.tdp , td.tdprule , td.tdP , td.tdPrule
1538     { padding: 0ex .5em 0ex .5em ; }
1539 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1540     { padding: 0ex .5em 0ex .5em ; }
1541 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1542     { padding: 0ex .5em 0ex .5em ; }
1543
1544
1545 /* table notes: */
1546 .tnotes {
1547     margin: 0ex 5% 1ex 5% ;
1548     padding: 0.5ex 1em 0.5ex 1em;
1549     font-size:.85em;
1550     text-align: left ;
1551 }
1552
1553 .tnotes dl dt p {margin-bottom:0px;}
1554
1555 .tnoteitemheader {margin-right: 1em;}
1556
1557
1558 /* for bigdelim */
1559 .ldelem, .rdelem { font-size: 200% }
1560
1561
1562 /* center, flushleft, flushright environments */
1563 div.center{text-align:center;}
1564 div.center table {margin-left:auto;margin-right:auto;}
1565 div.flushleft{text-align:left;}
1566 div.flushleft table {margin-left:0em ; margin-right:auto;}
1567 div.flushright{text-align:right;}
1568 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1569
1570
1571 /* Fancybox */
1572 div.Btrivlist table tr td { padding: .2ex 0em ; }
1573
1574
```

```
1575 /* program listing callouts: */
1576 span.callout {
1577     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1578             Geneva, Verdana, sans-serif ;
1579     border-radius: .5em;
1580     background-color:black;
1581     color:white;
1582     padding:0px .25em 0px .25em;
1583     margin: 0 ;
1584     font-weight: bold;
1585     font-size:.72em ;
1586 }
1587
1588 div.programlisting pre.verbatim span.callout{
1589     font-size: .85em ;
1590 }
1591
1592
1593
1594
1595
1596 div.published
1597 {
1598     text-align: center ;
1599     font-variant: normal ;
1600     font-style: italic ;
1601     font-size: 1em ;
1602     margin: 3ex 0em 3ex 0em ;
1603 }
1604
1605 div.subtitle
1606 {
1607     text-align: center ;
1608     font-variant: normal ;
1609     font-style: italic ;
1610     font-size: 1.25em ;
1611     margin: 3ex 0em 3ex 0em ;
1612 }
1613
1614 div.subtitle p { margin: 1ex ; }
1615
1616 div.author
1617 {
1618     font-variant: normal ;
1619     font-style: normal ;
1620     font-size: 1em ;
1621     margin: 3ex 0em 3ex 0em ;
1622 }
1623
1624 div.oneauthor {
```

```
1625     display: inline-block ;
1626     margin: 3ex 1em 0ex 1em ;
1627 }
1628
1629 /*
1630 div.author table {
1631     margin: 3ex auto 0ex auto ;
1632     background: none ;
1633 }
1634
1635 div.author table tbody tr td { padding: .25ex ; }
1636 */
1637
1638 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1639
1640 div.titledate {
1641     text-align: center ;
1642     font-size: .85em ;
1643     font-style: italic;
1644     margin: 6ex 0em 6ex 0em ;
1645 }
1646
1647
1648 nav.topnavigation{
1649     text-align: left ;
1650     padding: 0.5ex 1em 0.5ex 1em ;
1651 /*     margin: 2ex 0em 3ex 0em ; */
1652     margin: 0 ;
1653     border-bottom: 1px solid silver ;
1654     border-top: 1px solid silver ;
1655     clear:right ;
1656 }
1657
1658 nav.botnavigation{
1659     text-align: left ;
1660     padding: 0.5ex 1em 0.5ex 1em ;
1661 /*     margin: 3ex 0em 2ex 0em ; */
1662     margin: 0 ;
1663     border-top: 1px solid silver ;
1664     border-bottom: 1px solid silver ;
1665     clear:right ;
1666 }
1667
1668
1669 header{
1670     line-height: 1.2 ;
1671     font-size: 1em ;
1672 /*     border-bottom: 2px solid silver ; */
1673     margin: 0px ;
1674     padding: 0ex 1em 0ex 1em ;
```

```
1675     text-align:center ;
1676 }
1677
1678 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1679
1680
1681 footer{
1682     font-size: .85em ;
1683     line-height: 1.2 ;
1684     margin-top: 1ex ;
1685     border-top: 2px solid silver ;
1686     padding: 2ex 1em 2ex 1em ;
1687     clear:right ;
1688     text-align:left ;
1689 }
1690
1691
1692 a.linkhome { font-weight:bold ; font-size: 1em ;}
1693
1694
1695 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1696
1697 img.lateximage{
1698     padding: 0px 0px 0px 0px ;
1699     box-shadow: none ;
1700     border: none ;
1701     background: none ;
1702     margin: 0px 0px -.15ex 0px ;
1703         /* pdfcrop leaves a slight margin, adjust to baseline */
1704     max-width: 100% ;
1705     border-radius: 0ex ;
1706     border: none ;
1707 }
1708
1709
1710
1711 nav.sidetoc {
1712     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1713             "Lucida Bright", Georgia, serif;
1714     float:right ;
1715     width: 20%;
1716     border-left: 1px solid silver;
1717     border-top: 1px solid silver;
1718     border-bottom: 1px solid silver;
1719 /*     border-top: 2px solid #808080 ; */
1720     background: #FAF7F4 ;
1721     padding: 2ex 0em 2ex 1em ;
1722     margin: 0ex 0em 2ex 1em ;
1723     font-size:.9em ;
1724     border-radius: 20px 0px 0px 20px ;
```

```
1725      }
1726
1727 div.sidetoccontents {
1728 /*     border-top: 1px solid silver ; */
1729     overflow-y: auto ;
1730     width: 100% ;
1731     text-align: left ;
1732 }
1733
1734
1735 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1736     text-indent: 0 ; }
1737
1738 nav.sidetoc p a {color:black ; font-size: .7em ;}
1739
1740 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1741     border-bottom: 1px solid silver ;     }
1742
1743 nav.sidetoc a:hover {text-decoration: underline ; }
1744
1745
1746
1747 section.textbody { margin: 0ex 1em 0ex 1em ;}
1748
1749
1750 div.multicolsheading { -webkit-column-span: all;
1751     -moz-column-span: all; column-span: all; }
1752 div.multicols { -webkit-columns: 3 380px ;
1753     -moz-columns: 3 380px ; columns: 3 380px ; }
1754 div.multicols p {margin-top: 0ex}
1755
1756
1757
1758 /* Used to support algorithmicx: */
1759 span.floatright { float: right ; }
1760
1761
1762
1763
1764 /* Native LaTeX theorems: */
1765
1766 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1767 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1768
1769
1770 /* theorem, amsthm, and ntheorem packages */
1771
1772 span.theoremheader,
1773 span.theoremheaderplain,
1774 span.theoremheaderdefinition,
```

```
1775 span.theoremheaderbreak,
1776 span.theoremheadermarginbreak,
1777 span.theoremheaderchangebreak,
1778 span.theoremheaderchange,
1779 span.theoremheadermargin
1780 {
1781     font-style:normal ; font-weight: bold ; margin-right: 1em ;
1782 }
1783
1784 span.amsthmnameplain,
1785 span.amsthmnamedefinition,
1786 span.amsthmnumberplain,
1787 span.amsthmnumberdefinition
1788 {
1789     font-style:normal ; font-weight: bold ;
1790 }
1791
1792
1793 span.amsthmnameremark,
1794 span.amsthmnumberremark
1795 {font-style:italic ; font-weight: normal ; }
1796
1797
1798 span.amsthmnoteplain,
1799 span.amsthmnotedefinition
1800 {font-style:normal ;}
1801
1802
1803 span.theoremheaderremark,
1804 span.theoremheaderproof,
1805 span.amsthmproofname
1806 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1807
1808 span.theoremheadersc
1809 {
1810     font-style:normal ;
1811     font-variant: small-caps ;
1812     font-weight: normal ;
1813     margin-right: 1em ;
1814 }
1815
1816 .theoremendmark {float:right}
1817
1818 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1819 div.theorembodybreak, div.theorembodynonumberbreak,
1820 div.theorembodymarginbreak,
1821 div.theorembodychangebreak,
1822 div.theorembodychange,
1823 div.theorembodymargin
1824 {
```

```
1825     font-style:italic;
1826     margin-top: 3ex ; margin-bottom: 3ex ;
1827 }
1828
1829 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1830 div.theorembodyplainupright, nonumberplainuprightsc,
1831 div.amsthmbodydefinition, div.amsthmbodyremark,
1832 div.amsthmproof
1833 {
1834     font-style: normal ;
1835     margin-top: 3ex ; margin-bottom: 3ex ;
1836 }
1837
1838 span.amsthmnoteremark {}
1839
1840
1841
1842 /*
1843 For CSS LaTeX and related logos:
1844 Based on:
1845 http://edward.oconnor.cx/2007/08/tex-poshlet
1846 http://nitens.org/taraborelli/texlogo
1847 */
1848
1849 .latexlogofont {
1850     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1851             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1852     font-variant: normal ;
1853 }
1854
1855 .latexlogo {
1856     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1857             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1858     letter-spacing: .03em ;
1859     font-size: 1.1em;
1860 }
1861
1862 .latexlogo sup {
1863     text-transform: uppercase;
1864     letter-spacing: .03em ;
1865     font-size: 0.85em;
1866     vertical-align: 0.15em;
1867     margin-left: -0.36em;
1868     margin-right: -0.15em;
1869 }
1870
1871 .latexlogo sub {
1872     text-transform: uppercase;
1873     vertical-align: -0.5ex;
1874     margin-left: -0.1667em;
```

```
1875   margin-right: -0.125em;
1876   font-size: 1em;
1877 }
1878
1879 .xetexlogo {
1880   font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1881   "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1882   letter-spacing: .03em ;
1883   font-size: 1.1em;
1884 }
1885
1886 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1887 .xetexlogo sub {
1888   text-transform: uppercase;
1889   vertical-align: -0.5ex;
1890   margin-left: -0.0667em;
1891   margin-right: -0.2em;
1892   font-size: 1em;
1893   letter-spacing: .03em ;
1894 }
1895
1896 /* A large gap between Xe and LaTeX v.s. TeX: */
1897 .xelatexlogo sub {
1898   text-transform: uppercase;
1899   vertical-align: -0.5ex;
1900   margin-left: -0.0667em;
1901   margin-right: -.05em;
1902   font-size: 1em;
1903   letter-spacing: .03em ;
1904 }
1905
1906 .amslogo {
1907   font-family: "TeXGyreChorus","URW Chancery L",
1908   "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
1909   "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1910   "Hoefler Text", Times, "Times New Roman", serif;
1911   font-style: italic;
1912 }
1913
1914 .lyxlogo {
1915   font-family: "URW Classico", Optima, "Linux Biolinum O",
1916   "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1917   Verdana, sans-serif ;
1918 }
1919
1920
1921
1922
1923 /* Only display top and bottom navigation if a small screen: */
1924 /* Hide the sidetoc if a small screen: */
```

```
1925 nav.topnavigation { display:none; }
1926 nav.botnavigation { display:none; }
1927
1928 @media screen and (max-width: 45em) {
1929 /*      nav.sidetoc {display:none;} */
1930     nav.sidetoc {
1931         float: none ;
1932         width: 100% ;
1933         margin: 5ex 0px 5ex 0px ;
1934         padding: 0 ;
1935         border-radius: 0 ;
1936         border-bottom: 1px solid black ;
1937         border-top: 1px solid black ;
1938         box-shadow: none ;
1939     }
1940 /*      nav.topnavigation { display:block } */
1941     nav.botnavigation { display:block }
1942     .marginpar {
1943         max-width: 100%;
1944         float: none;
1945         display:block ;
1946         margin: 1ex 1em 1ex 1em ;
1947     }
1948 }
1949
1950 @media print {
1951     body {
1952         font-family: "Linux Libertine O",
1953             "DejaVu Serif", "Bitstream Vera Serif",
1954             "Liberation Serif", "Nimbus Roman No 9 L",
1955             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1956     }
1957     nav.sidetoc { display:none; }
1958     nav.topnavigation { display: none; }
1959     nav.botnavigation { display: none; }
1960 }
1961
1962 @media handheld {
1963     nav.sidetoc { display:none; }
1964     nav.topnavigation { display:block }
1965     nav.botnavigation { display:block }
1966 }
1967
1968 @media projection {
1969     nav.sidetoc { display:none; }
1970     nav.topnavigation { display:block }
1971     nav.botnavigation { display:block }
1972 }
1973 \end{filecontents*}
1974 % \end{Verbatim}% for syntax highlighting
```

```
1975 \end{warpprint}
```

30.5 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.

```
1976 \begin{warpprint}
1977 \begin{filecontents*}{lwarp_sagebrush.css}
1978 @import url("lwarp.css") ;
1979
1980
1981 A:link {color:#105030 ; text-decoration: none ; }
1982 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
1983 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
1984 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
1985
1986
1987
1988 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1989 {
1990     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1991             "Linux Libertine 0", "Liberation Serif",
1992             "Nimbus Roman No 9 L", "FreeSerif",
1993             "Hoefler Text", Times, "Times New Roman", serif;
1994     font-variant: small-caps ;
1995     font-weight: normal ;
1996     color: #304070 ;
1997     text-shadow: 2px 2px 3px #808080;
1998 }
1999
2000 h1 { /* title of the entire website, used on each page */
2001     font-variant: small-caps ;
2002     color: #304070 ;
2003     text-shadow: 2px 2px 3px #808080;
2004     background-color: #F7F7F0 ;
2005     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2006 }
2007
2008 h1 {
2009     border-bottom: 1px solid #304070;
2010     border-top: 2px solid #304070;
2011 }
2012
2013 h2 {
```

```
2014 border-bottom: 1px solid #304070;
2015 border-top: 2px solid #304070;
2016 background-color: #F7F7F0 ;
2017 background-image: linear-gradient(to bottom, #F7F7F0, #DADOC0);
2018 }
2019
2020
2021
2022 div.abstract {
2023     background: #f5f5eb ;
2024     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2025
2026     border: 1px solid silver;
2027     border-radius: 1em ;
2028 }
2029
2030 div.abstract dl {line-height:1.5;}
2031 div.abstract dt {color:#304070;}
2032
2033 div.abstracttitle{
2034     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2035         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2036         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2037     font-weight:bold;
2038     font-variant: small-caps ;
2039     font-size:1.5em;
2040     border-bottom: 1px solid silver ;
2041     color: #304070 ;
2042     text-align: center ;
2043     text-shadow: 1px 1px 2px #808080;
2044 }
2045
2046 span.abstractrunintitle{
2047     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2048         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2049         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2050     font-weight:bold;
2051 }
2052
2053
2054 div.epigraph {
2055     background: #f5f5eb ;
2056     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2057
2058     border: 1px solid silver ;
2059     border-radius: 1ex ;
2060     box-shadow: 3px 3px 3px #808080 ;
2061 }
2062
2063
```

```
2064 .example {
2065     background-color: #f5f5eb ;
2066     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2067
2068 }
2069
2070 div.exampletitle{
2071     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2072         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2073         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2074     font-weight:bold;
2075     font-variant: small-caps ;
2076     border-bottom: 1px solid silver ;
2077     color: #304070 ;
2078     text-align: center ;
2079     text-shadow: 1px 1px 2px #808080;
2080 }
2081
2082
2083 .sidebar {
2084     background-color: #f5f5eb ;
2085     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2086
2087 }
2088
2089 div.sidebartitle{
2090     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2091         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2092         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2093     font-weight:bold;
2094     font-variant: small-caps ;
2095     border-bottom: 1px solid silver ;
2096     color: #304070 ;
2097     text-align: center ;
2098     text-shadow: 1px 1px 2px #808080;
2099 }
2100
2101
2102 .fancyvrblabel {
2103     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2104         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2105         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2106     font-weight:bold;
2107     font-variant: small-caps ;
2108     font-size: 1.5em ;
2109     color: #304070 ;
2110     text-align: center ;
2111     text-shadow: 1px 1px 2px #808080;
2112 }
2113
```

```
2114
2115
2116 div.minipage {
2117     background-color: #eeeeee7 ;
2118     border: 1px solid silver ;
2119     border-radius: 1ex ;
2120 }
2121
2122 section.textbody > div.minipage {
2123     box-shadow: 3px 3px 3px #808080 ;
2124 }
2125
2126 div.fboxBlock div.minipage { box-shadow: none ; }
2127
2128 .framed .minipage , .framedleftbar .minipage {
2129     border: none ;
2130     background: none ;
2131     padding: 0ex ;
2132     margin: 0ex ;
2133 }
2134
2135 figure.figure .minipage, figcaption .minipage { border: none; }
2136
2137 div.marginblock div.minipage { border: none; }
2138
2139 figure , div.marginblock {
2140     background-color: #eeeeee7 ;
2141     border: 1px solid silver ;
2142     border-radius: 1ex ;
2143     box-shadow: 3px 3px 3px #808080 ;
2144 }
2145
2146 figure figure {
2147     border: 1px solid silver ;
2148     margin: 0em ;
2149     box-shadow: none ;
2150 }
2151
2152 /*
2153 figcaption {
2154     border-top: 1px solid silver ;
2155     border-bottom: 1px solid silver ;
2156     background-color: #e8e8e8 ;
2157 }
2158 */
2159
2160
2161 div.table {
2162     box-shadow: 3px 3px 3px #808080 ;
2163 }
```

```
2164
2165 /*
2166 .tnotes {
2167     background: #e8e8e8;
2168     border: 1px solid silver;
2169 }
2170 */
2171
2172
2173 nav.topnavigation{
2174     background-color: #b0b8b0 ;
2175     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2176 }
2177
2178 nav.botnavigation{
2179     background-color: #b0b8b0 ;
2180     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2181 }
2182
2183
2184
2185 header{
2186     background-color: #F7F7F0 ;
2187     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2188 }
2189
2190 footer{
2191     background-color: #F7F7F0 ;
2192     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2193 }
2194
2195
2196
2197 nav.sidetoc {
2198     background-color: #F7F7F0 ;
2199     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2200     box-shadow: 3px 3px 3px #808080 ;
2201     border-radius: 0px 0px 0px 20px ;
2202 }
2203
2204 div.sidetotitle {color: #304070 ; }
2205
2206 nav.sidetoc a:hover {
2207     color:#006000 ;
2208     text-decoration: none ;
2209     text-shadow:0px 0px 2px #a0a0a0;
2210 }
2211
2212
2213 @media screen and (max-width: 45em) {
```

```
2214     nav.sidetoc { border-radius: 0 ; }
2215 }
2216
2217
2218 \end{filecontents*}
2219 % \end{Verbatim}%
2220 \end{warpprint}
```

30.6 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.

```
2221 \begin{warpprint}
2222 \begin{filecontents*}{lwarp_formal.css}
2223 @import url("lwarp.css") ;
2224
2225
2226
2227 A:link {color:#802020 ; text-decoration:none; }
2228 A:visited {color:#802020 ; text-shadow:none ;}
2229 A:hover {color:#400000 ; text-shadow:none ;}
2230 A:active {color:#C00000 ; text-shadow:none ;}
2231
2232
2233 body {
2234     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2235         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2236         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2237         "Times New Roman", serif;
2238     background: #ffffcf5;
2239 }
2240
2241 span.textrm {
2242     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2243         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2244         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2245         "Times New Roman", serif;
2246 }
2247
2248 span.textsf {
2249     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2250         Geneva, Verdana, sans-serif ;
2251 }
2252
```

```
2253
2254
2255 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2256 {
2257     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2258         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2259         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2260         "Times New Roman", serif;
2261     color: #800000 ;
2262     text-shadow: none ;
2263 }
2264
2265 h1, h2 {
2266     background-color: #ffffcf5 ;
2267     background-image: none ;
2268     border-bottom: 1px solid #808080;
2269     border-top: 2px solid #808080;
2270 }
2271
2272 div.abstracttitle {
2273     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2274         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2275         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2276         "Times New Roman", serif;
2277     color: black ;
2278     text-shadow: none ;
2279 }
2280
2281 span.abstractrunintitle {
2282     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2283         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2284         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2285         "Times New Roman", serif;
2286     color: black ;
2287     text-shadow: none ;
2288 }
2289
2290 div.abstract { font-size: 100% }
2291
2292 .sidebar {
2293     background: #ffffcf5;
2294     background-image: none ;
2295     margin: 2em 5% 2em 5%;
2296     padding: 0.5em 1em;
2297     border: none ;
2298     border-top : 1px solid silver;
2299     border-bottom : 1px solid silver;
2300     font-size: 90% ;
2301 }
2302
```

```
2303 div.sidebartitle{  
2304     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",  
2305         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",  
2306         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,  
2307         "Times New Roman", serif;  
2308     color: #800000 ;  
2309     text-shadow: none ;  
2310     border: none ;  
2311 }  
2312  
2313 .example {  
2314     background: #fffcf5;  
2315     background-image: none ;  
2316     margin: 2em 5% 2em 5%;  
2317     padding: 0.5em 1em;  
2318     border: none ;  
2319     border-top : 1px solid silver;  
2320     border-bottom : 1px solid silver;  
2321 }  
2322  
2323 div.exampletitle{  
2324     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",  
2325         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",  
2326         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,  
2327         "Times New Roman", serif;  
2328     color: #800000 ;  
2329     text-shadow: none ;  
2330     border: none ;  
2331 }  
2332  
2333 div.fancyvrblabel{  
2334     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",  
2335         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",  
2336         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,  
2337         "Times New Roman", serif;  
2338     color: #800000 ;  
2339     text-shadow: none ;  
2340     border: none ;  
2341 }  
2342  
2343  
2344  
2345 .verse {  
2346     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",  
2347         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",  
2348         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,  
2349         "Times New Roman", serif;  
2350 }  
2351  
2352
```

```
2353 figure {
2354     margin: 3ex 5% 3ex 5% ;
2355     padding: 1ex 1em 1ex 1em ;
2356     background-color: #ffffcf5 ;
2357     overflow-x: auto ;
2358     border: none ;
2359 /*      border-top: 1px solid silver; */
2360 /*      border-bottom: 1px solid silver; */
2361 }
2362
2363
2364 figcaption , .lstlisting {
2365     border: none ;
2366 /*      border-top: 1px solid silver ; */
2367 /*      border-bottom: 1px solid silver ; */
2368     background-color: #ffffcf5 ;
2369 }
2370
2371 .tnotes {
2372     background: #ffffcf5 ;
2373 }
2374
2375 .theorem {
2376     background: none ;
2377 }
2378
2379 .minipage {
2380     background-color: #ffffcf5 ;
2381     border: none ;
2382 }
2383
2384 div.floatrow figure { border: none ; }
2385
2386 figure figure { border: none ; }
2387
2388
2389 nav.toc, nav.lof, nav.lot, nav.loi {
2390     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2391         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2392         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2393         "Times New Roman", serif;
2394 }
2395
2396 nav.sidetoc {
2397     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2398         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2399         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2400         "Times New Roman", serif;
2401     background-image: linear-gradient(to bottom, #ffffcf5, #C0C0C0);
2402     border-radius: 0px 0px 0px 20px ;
```

```
2403 }
2404
2405 div.sidetotoctitle{
2406     color: #800000 ;
2407 }
2408
2409 header{
2410     background-color: #e0e0e0 ;
2411     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2412     text-align:center ;
2413 }
2414
2415 footer{
2416     background-color: #e0e0e0 ;
2417     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
2418     padding: 2ex 1em 2ex 1em ;
2419     clear:right ;
2420     text-align:left ;
2421 }
2422
2423 nav.botnavigation {
2424     background: #dedcd5 ;
2425     border-top: 1px solid black ;
2426 }
2427 \end{filecontents*}
2428 % \end{Verbatim}%
2429 \end{warpprint}
```

30.7 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.

```
2430 \begin{warpprint}
2431 \begin{filecontents*}{sample_project.css}
2432 /* ( --- Start of project.css --- ) */
2433 /* A sample project-specific CSS file for lwarp --- ) */
2434
2435 /* Load default lwarp settings: */
2436 @import url("lwarp.css") ;
2437 /* or lwarp_formal.css, lwarp_sagebrush.css */
2438
2439 /* Project-specific CSS setting follow here. */
2440 /* . . . */
2441
```

```

2442 /* ( --- End of project.css --- ) */
2443 \end{filecontents*}
2444 % \end{Verbatim} for syntax highlighting
2445 \end{warpprint}
```

30.8 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.

```

2446 \begin{warpprint}
2447 \begin{filecontents*}{lwarp.xdy}
2448 (require "tex/inputenc/latin.xdy")
2449 (merge-rule "\PS *" "Postscript")
2450 (require "texindy.xdy")
2451 (require "page-ranges.xdy")
2452 (require "book-order.xdy")
2453 (markup-locref :open "\hyperindexref{" :close "}")
2454 \end{filecontents*}
2455 % \end{Verbatim} for syntax highlighting
2456 \end{warpprint}
```

30.9 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.

```

2457 \begin{warpprint}
2458 \begin{filecontents*}{lwarp_mathjax.txt}
2459 <!-- https://groups.google.com/forum/#!topic/
2460           mathjax-users/jUtewUcE2bY -->
2461 <script type="text/x-mathjax-config">
2462 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2463   var seteqsectionDefault = {name: "", num: 0};
2464   var seteqsections = {}, seteqsection = seteqsectionDefault;
2465   var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2466   var AMS = MathJax.Extension["TeX/AMSmash"];
2467   TEX.Definitions.Add({
2468     macros: {
2469       seteqsection: "mySection",
2470       seteqnumber: "mySetEqNumber"
```

```
2471      }
2472  });
2473
2474  PARSE.Augment({
2475    mySection: function (name) {
2476      seteqsection.num = AMS.number;
2477      var n = this.GetArgument(name);
2478      if (n === "") {
2479        seteqsection = seteqsectionDefault;
2480      } else {
2481        if (!seteqsections["_"+n])
2482          seteqsections["_"+n] = {name:n, num:0};
2483        seteqsection = seteqsections["_"+n];
2484      }
2485      AMS.number = seteqsection.num;
2486    },
2487    mySetEqNumber: function (name) {
2488      var n = this.GetArgument(name);
2489      if (!n || !n.match(/^\ *[0-9]+ *\$/))
2490        n = ""; else n = parseInt(n)-1;
2491      <!-- $ syntax highlighting -->
2492      if (n === "" || n < 1)
2493        TEX.Error
2494          ("Argument to "+name+" should be a positive integer");
2495      AMS.number = n;
2496    }
2497  });
2498  MathJax.Hub.Config({
2499    TeX: {
2500      equationNumbers: {
2501        formatTag: function (n)
2502          {return "("+(seteqsection.name+"."+n).replace(/\.\./,"")+")"},

2503        formatID: function (n) {
2504          n = (seteqsection.name+'.'+n).replace
2505            (/[:'<>&]/g,"").replace(/\.\./,"");
2506          return 'mjax-eqn-' + n;
2507        }
2508      }
2509    }
2510  });
2511 });
2512 </script>
2513
2514 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2515 <script type="text/x-mathjax-config">
2516   MathJax.Ajax.config.path["Contrib"] =
2517     "https://cdn.mathjax.org/mathjax/contrib";
2518 </script>
2519
2520 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
```

```

2521                                         tree/master/siunitx -->
2522 <script type="text/x-mathjax-config">
2523   MathJax.Hub.Config({
2524     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
2525     jax: ["input/TeX", "output/HTML-CSS"],
2526     tex2jax: {inlineMath: [[["$","$"], ["\\(", "\\)"]]]},
2527     TeX: {extensions: ["AMSmath.js", "AMSSymbols.js", "sinuitx.js"]}
2528   });
2529 </script>
2530
2531 <script type="text/x-mathjax-config">
2532 MathJax.Hub.Config({
2533   TeX: {
2534     equationNumbers: {
2535       autoNumber: "AMS"
2536     }
2537   }
2538 });
2539 </script>
2540
2541 <!-- Alternative CDN provider: -->
2542 <script type="text/javascript" async
2543 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2544 </script>
2545
2546 <!-- No longer supported after April 30, 2017: -->
2547 <!--
2548 <script
2549   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2550 </script>
2551 -->
2552
2553 \end{filecontents*}
2554 % \end{Verbatim}%
2555 \end{warpprint}

```

30.10 lwarfmk option

The following is only generated if the `lwarfmk` option was given to `lwarf`.

```
2556 \begin{LWR@createlwarfmk}
```

Prog `lwarfmk` Creates a local copy of `lwarfmk`:

```

2557 \begin{filecontents*}{lwarfmk.lua}
2558 #!/usr/bin/env texlua
2559

```

```
2560 -- Copyright 2016-2017 Brian Dunn
2561
2562 -- Print the usage of the lwarpmk command:
2563
2564 printversion = "v0.39"
2565
2566 function printhelp ()
2567 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2568 end
2569
2570 function printusage ()
2571 print ( [[
2572
2573 lwarpmk print [project]: Compile the print version if necessary.
2574 lwarpmk print1 [project]: Forced single compile of the print version.
2575 lwarpmk printindex [project]: Process the index for the print version.
2576 lwarpmk printglossary [project]: Process the glossary for the print version.
2577 lwarpmk html [project]: Compile the HTML version if necessary.
2578 lwarpmk html1 [project]: Forced single compile of the HTML version.
2579 lwarpmk htmlindex [project]: Process the index for the html version.
2580 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2581 lwarpmk again [project]: Touch the source code to trigger recompiles.
2582 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2583 lwarpmk pdftohtml [project]:
2584     For use with latexmk or a Makefile:
2585     Convert project_html.pdf to project_html.html and
2586     individual HTML files.
2587 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*., .gl*
2588 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2589 lwarpmk -h: Print this help message.
2590 lwarpmk --help: Print this help message.
2591
2592 ]] )
2593 printconf ()
2594 end
2595
2596 -- Print the format of the configuration file lwarpmk.conf:
2597
2598 function printconf ()
2599 print ( [[
2600 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2601 --
2602 opsystem = "Unix" (or "Windows")
2603 latexname = "pdflatex" (or "lualatex", or "xelatex")
2604 sourcename = "projectname" (the source-code filename w/o .tex)
2605 homehtmlfilename = "index" (or perhaps the project name)
2606 htmlfilename = "" (or "projectname" - filename prefix)
2607 latexmk = "false" (or "true" to use latexmk to build PDFs)
2608 languge = "english" (use a language supported by xindy)
2609 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
```

```
2610 --
2611 Filenames must contain only letters, numbers, underscore, or dash.
2612 Values must be in "quotes".
2613
2614 ]] ) ;
2615 end
2616
2617
2618 -- Split one large sourcefile into a number of files,
2619 -- starting with destfile.
2620 -- The file is split at each occurance of <!--|Start file|newfilename|*
2621
2622 function splitfile (destfile,sourcefile)
2623 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2624 local sfile = io.open(sourcefile)
2625 io.output(destfile)
2626 for line in sfile:lines() do
2627 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
2628 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2629 io.output(newfilename) ;
2630 else -- not a splitpoint
2631 io.write (line .. "\n") ;
2632 end
2633 end -- do
2634 io.close(sfile)
2635 end -- function
2636
2637 -- Incorrect value, so print an error and exit.
2638
2639 function cvalueerror ( line, linenum , cvalue )
2640     print ( linenum .. " : " .. line ) ;
2641     print ("lwarpmk: incorrect variable value \" .. cvalue .. \" in lwarpmk.conf.\n" ) ;
2642     printconf () ;
2643     os.exit(1) ;
2644 end
2645
2646 -- Load settings from the project's "lwarpmk.conf" file:
2647
2648 function loadconf ()
2649 -- Default configuration filename:
2650 local conffile = "lwarpmk.conf"
2651 -- Optional configuration filename:
2652 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2653 -- Default language:
2654 language = "english"
2655 -- Default xdyfile:
2656 xdyfile = "lwarp.xdy"
2657 -- Verify the file exists:
2658 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2659 print("lwarpmk: " .. conffile .." does not exist.")
```

```
2660 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")  
2661 printhelp () ;  
2662 os.exit(1) -- exit the entire lwarpmk script  
2663 else -- file exists  
2664 -- Read the file:  
2665 print ("lwarpmk: Reading " .. conffile ..".")  
2666 local cfile = io.open(conffile)  
2667 -- Scan each line:  
2668 local linenum = 0  
2669 for line in cfile:lines() do -- scan lines  
2670 linenum = linenum + 1  
2671 i,j,cvarname,cvalue = string.find (line,"(%w-[_]*%)%s*=%s*\"(%w%-_.*)\"") ;  
2672 -- Error if incorrect enclosing characters:  
2673 if ( i == nil ) then  
2674 print ( linenum .. " : " .. line ) ;  
2675 print ( "lwarpmk: Incorrect entry in " .. conffile ..".\n" ) ;  
2676 printconf () ;  
2677 os.exit(1) ;  
2678 end  
2679 if ( cvarname == "opsystem" ) then  
2680     -- Verify choice of opsystem:  
2681     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then  
2682         opsystem = cvalue  
2683     else  
2684         cvalueerror ( line, linenum , cvalue )  
2685     end  
2686 elseif ( cvarname == "latexname" ) then  
2687     -- Verify choice of LaTeX compiler:  
2688     if (  
2689         (cvalue == "pdflatex") or  
2690         (cvalue == "xelatex") or  
2691         (cvalue == "lualatex")  
2692     ) then  
2693         latexname = cvalue  
2694     else  
2695         cvalueerror ( line, linenum , cvalue )  
2696     end  
2697 elseif ( cvarname == "sourcename" ) then sourcename = cvalue  
2698 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue  
2699 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue  
2700 elseif ( cvarname == "texmk" ) then texmk = cvalue  
2701 elseif ( cvarname == "language" ) then language = cvalue  
2702 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue  
2703 else  
2704 print ( linenum .. " : " .. line ) ;  
2705 print ( "lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile ..".\n" ) ;  
2706 printconf () ;  
2707 os.exit(1) ;  
2708 end  
2709 end -- do scan lines
```

```
2710 io.close(cfile)
2711 end -- file exists
2712 -- Select some operating-system commands:
2713 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2714 rmname = "rm"
2715 mvname = "mv"
2716 touchnamepre = "touch"
2717 touchnamepost = ""
2718 dirslash = "/"
2719 opquote= "\'"
2720 elseif opsystem=="Windows" then -- For Windows
2721 rmname = "DEL"
2722 mvname = "MOVE"
2723 touchnamepre = "COPY /b"
2724 touchnamepost = "+,,"
2725 dirslash = "\\"
2726 opquote= "\""
2727 else print ( "lwarfpmk: Select Unix or Windows for opsystem" )
2728 end --- for Windows
2729
2730 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2731 if ( latexname == "pdflatex" ) then
2732 xindycmd = "texindy -C utf8"
2733 glossarycmd = "xindy -C utf8"
2734 else
2735 xindycmd = "xindy -M texindy -C utf8"
2736 glossarycmd = "xindy -C utf8"
2737 end
2738
2739 end -- loadconf
2740
2741
2742 function refreshdate ()
2743 os.execute(touchnamepre .. " " .. sourcename .. ".tex" .. touchnamepost)
2744 end
2745
2746
2747 -- Scan the LaTeX log file for the phrase "Rerun to get",
2748 -- indicating that the file should be compiled again.
2749 -- Return true if found.
2750
2751 function reruntoget (filesource)
2752 local fsource = io.open(filesource)
2753 for line in fsource:lines() do
2754 if ( string.find(line,"Rerun to get") ~= nil ) then
2755     io.close(fsource)
2756     return true
2757 end
2758 end
2759 io.close(fsource)
```

```
2760 return false
2761 end
2762
2763
2764 -- Compile one time, return true if should compile again.
2765 -- fsuffix is "" for print, "_html" for HTML output.
2766
2767 function onetime (fsuffix)
2768 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2769 err = os.execute(
2770 --     "echo " ..
2771     latexname .. " " .. sourcename..fsuffix )
2772 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2773 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2774 end
2775
2776
2777 -- Compile up to five times.
2778 -- fsuffix is "" for print, "_html" for HTML output
2779
2780 function manytimes (fsuffix)
2781 if onetime(fsuffix) == true then
2782 if onetime(fsuffix) == true then
2783 if onetime(fsuffix) == true then
2784 if onetime(fsuffix) == true then
2785 if onetime(fsuffix) == true then
2786 end end end end end
2787 end
2788
2789 -- Exit if the given file does not exist.
2790
2791 function verifyfileexists (filename)
2792 if (lfs.attributes ( filename , "modification" ) == nil ) then
2793 print ( "lwarpmk: " .. filename .. " not found." ) ;
2794 os.exit (1) ;
2795 end
2796 end
2797
2798
2799 -- Convert <project>_html.pdf into HTML files:
2800
2801 function pdftohtml ()
2802     -- Convert to text:
2803     print ("lwarpmk: Converting " .. sourcename
2804             .."_html.pdf to " .. sourcename .. "_html.html")
2805     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2806               .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2807     -- Split the result into individual HTML files:
2808     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2809 end
```

```
2810
2811
2812 -- Remove auxiliary files:
2813
2814 function removeaux ()
2815     os.execute ( rmname .. " " ..
2816         sourcename .. ".aux" .. sourcename .. "_html.aux" ..
2817         sourcename .. ".toc" .. sourcename .. "_html.toc" ..
2818         sourcename .. ".lof" .. sourcename .. "_html.lof" ..
2819         sourcename .. ".lot" .. sourcename .. "_html.lot" ..
2820         sourcename .. ".idx" .. sourcename .. "_html.idx" ..
2821         sourcename .. ".ind" .. sourcename .. "_html.ind" ..
2822         sourcename .. ".log" .. sourcename .. "_html.log" ..
2823         sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
2824         "*_html_inc.*"
2825     )
2826 end
2827
2828
2829
2830 -- Create lateximages based on lateximages.txt:
2831 function createlateximages ()
2832 print ("lwarpmk: Creating lateximages.")
2833 local limagesfile = io.open("lateximages.txt")
2834 -- Create the lateximages directory, ignore error if already exists
2835 err = os.execute("mkdir lateximages")
2836 -- Scan lateximages.txt
2837 for line in limagesfile:lines() do
2838 -- lwimgpage is the page number in the PDF which has the image
2839 -- lwimgnum is the sequential lateximage number to assign for the image
2840 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
2841 -- For each entry:
2842 if ( (i~=nil) ) then
2843 -- Separate out the image into its own single-page pdf:
2844 err = os.execute(
2845 "pdfseparate -f" .. lwimgpage .. " -l" ..
2846 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2847 -- Crop the image:
2848 err = os.execute(
2849 "pdfcrop --hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2850 if ( err ~= 0 ) then print ("lwarpmk: File error.") ; os.exit(1) ; end
2851 -- Convert the image to svg:
2852 err = os.execute(
2853 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2854 if ( err ~= 0 ) then print ("lwarpmk: File error.") ; os.exit(1) ; end
2855 -- Move the result into lateximages/:
2856 err = os.execute(
2857 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2858 if ( err ~= 0 ) then print ("lwarpmk: File error.") ; os.exit(1) ; end
2859 -- Remove the temporary files:
```

```
2860 err = os.execute(
2861 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2862 if ( err ~= 0 ) then print ( "lwarfpmk: File error." ) ; os.exit(1) ; end
2863 end
2864 end -- do
2865 io.close(limagesfile)
2866 end -- function
2867
2868
2869 -- Use latexmk to compile source and index:
2870 -- fsuffix is "" for print, or "_html" for HTML
2871 function compilelatexmk ( fsuffix )
2872     -- The recorder option is required to detect changes in <project>.tex
2873     -- while we are loading <project>_html.tex.
2874     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2875         .. "-e "
2876         .. opquote
2877         .. "$makeindex = q/" -- $
2878         .. xindycmd
2879         .. " -M " .. xdyfile
2880         .. " -L " .. language .. " /"
2881         .. opquote
2882         .. " -pdflatex="" .. latexname .. "%0 %S\" "
2883         .. sourcename..fsuffix .. ".tex" ) ;
2884     if ( err ~= 0 ) then print ( "lwarfpmk: Compile error." ) ; os.exit(1) ; end
2885 end
2886
2887
2888
2889 -- lwarfpmk --version :
2890
2891 if (arg[1] == "--version") then
2892 print ( "lwarfpmk: " .. printversion )
2893
2894 else -- not --version
2895
2896 -- print intro:
2897
2898 print ("lwarfpmk: " .. printversion .. " Automated make for the LaTeX lwarf package.")
2899
2900 -- lwarfpmk print:
2901
2902 if arg[1] == "print" then
2903 loadconf ()
2904 if ( latexmk == "true" ) then
2905     compilelatexmk ("")
2906     print ("lwarfpmk: Done.")
2907 else -- not latexmk
2908     verifyfileexists (sourcename .. ".tex") ;
2909     -- See if up to date:
```

```
2910     if (
2911         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2912         (
2913             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2914             lfs.attributes ( sourcename .. ".pdf" , "modification" )
2915         )
2916     ) then
2917         -- Recompile if not yet up to date:
2918         manytimes("")
2919         print ("lwarpmk: Done.") ;
2920     else
2921         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
2922     end
2923 end -- not latexmk
2924
2925 elseif arg[1] == "print1" then
2926     loadconf ()
2927     verifyfileexists (sourcename .. ".tex") ;
2928     onetime("")
2929     print ("lwarpmk: Done.") ;
2930
2931 -- lwarf printindex:
2932 -- Compile the index then touch the source
2933 -- to trigger a recompile of the document:
2934
2935 elseif arg[1] == "printindex" then
2936 loadconf ()
2937 print ("lwarpmk: Processing the index.")
2938 os.execute(
2939     xindycmd
2940     .. " -M " .. xdyfile
2941     .. " -L " .. language
2942     .. " " .. sourcename .. ".idx")
2943 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2944 refreshdate ()
2945 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2946 print ("lwarpmk: Done.")
2947
2948 -- lwarf printglossary:
2949 -- Compile the glossary then touch the source
2950 -- to trigger a recompile of the document:
2951
2952 elseif arg[1] == "printglossary" then
2953 loadconf ()
2954 print ("lwarpmk: Processing the glossary.")
2955
2956 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2957     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2958     .. sourcename .. ".glo")
2959 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
```

```
2960 refreshdate ()
2961 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2962 print ("lwarpmk: Done.")
2963
2964 -- lwarpmk html:
2965
2966 elseif arg[1] == "html" then
2967 loadconf ()
2968 if ( latexmk == "true" ) then
2969     compilelatexmk ("_html")
2970     pdftohtml ()
2971     print ("lwarpmk: Done.")
2972 else -- not latexmk
2973     verifyfileexists ( sourcename .. ".tex" ) ;
2974     -- See if exists and is up to date:
2975     if (
2976         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
2977         (
2978             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2979             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
2980         )
2981     ) then
2982         -- Recompile if not yet up to date:
2983         manytimes("_html")
2984         pdftohtml ()
2985         print ("lwarpmk: Done.")
2986     else
2987         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
2988     end
2989 end -- not latexmk
2990
2991 elseif arg[1] == "html1" then
2992     loadconf ()
2993     verifyfileexists ( sourcename .. ".tex" ) ;
2994     onetime("_html")
2995     pdftohtml ()
2996     print ("lwarpmk: Done.")
2997
2998 elseif arg[1] == "pdftohtml" then
2999     loadconf ()
3000     pdftohtml ()
3001
3002 -- lwarpmk htmlindex:
3003 -- Compile the index then touch the source
3004 -- to trigger a recompile of the document:
3005
3006 elseif arg[1] == "htmlindex" then
3007 loadconf ()
3008 print ("lwarpmk: Processing the index.")
3009 os.execute(
```

```
3010     xindycmd
3011     .. " -M " .. xdyfile
3012     .. " -L " .. language
3013     .. " .. sourcename .. "_html.idx"
3014 )
3015 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3016 refreshdate ()
3017 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3018 print ("lwarpmk: Done.")
3019
3020 -- lwarpmk htmlglossary:
3021 -- Compile the glossary then touch the source
3022 -- to trigger a recompile of the document:
3023
3024 elseif arg[1] == "htmlglossary" then
3025 loadconf ()
3026 print ("lwarpmk: Processing the glossary.")
3027
3028 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3029     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3030     "_html.gls " .. sourcename .. "_html.glo")
3031
3032 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3033 refreshdate ()
3034 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3035 print ("lwarpmk: Done.")
3036
3037 -- lwarpmk limages:
3038 -- Scan the lateximages.txt file to create lateximages,
3039 -- then touch the source to trigger a recompile.
3040
3041 elseif arg[1] == "limages" then
3042 loadconf ()
3043 print ("lwarpmk: Processing images.")
3044 createlateximages ()
3045 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3046 refreshdate ()
3047 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3048 print ("lwarpmk: Done.")
3049
3050 -- lwarpmk again:
3051 -- Touch the source to trigger a recompile.
3052
3053 elseif arg[1] == "again" then
3054 loadconf ()
3055 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3056 refreshdate ()
3057 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3058 print ("lwarpmk: Done.")
3059
```

```
3060 -- lwarpmk clean:  
3061 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.* , .gl*  
3062  
3063 elseif arg[1] == "clean" then  
3064 loadconf ()  
3065 removeaux ()  
3066 print ("lwarpmk: Done.")  
3067  
3068 -- lwarpmk cleanall  
3069 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.* , .gl*  
3070 --     and also project.pdf, *.html  
3071  
3072 elseif arg[1] == "cleanall" then  
3073 loadconf ()  
3074 removeaux ()  
3075 os.execute ( rmname .. " " ..  
3076     sourcename .. ".pdf" .. sourcename .. "_html.pdf" ..  
3077     "*_.html"  
3078     )  
3079 print ("lwarpmk: Done.")  
3080  
3081 -- lwarpmk with no argument :  
3082  
3083 elseif (arg[1] == nil) then  
3084 printhelp ()  
3085  
3086 -- lwarpmk -h or lwarpmk --help :  
3087  
3088 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then  
3089 printusage ()  
3090  
3091 else  
3092 print ("lwarpmk: Unknown command \"\"..arg[1].."\".\n")  
3093 printhelp ()  
3094 end  
3095  
3096 end -- not --version  
3097 \end{filecontents*}  
3098 % \end{Verbatim} % for syntax highlighting  
  
3099 \end{LWR@createlwarpmk}
```

31 Stacks

for HTML output: 3100 \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.

31.1 Assigning depths

initial depths for empty stack entries:

```
3101 \newcommand*{\LWR@depthnone}{-5}
```

all sectioning depths are deeper than \LWR@depthfinished:

```
3102 \newcommand*{\LWR@depthfinished}{-4}
3103 \newcommand*{\LWR@depthpart}{-1}
3104 \newcommand*{\LWR@depthchapter}{0}
3105 \newcommand*{\LWR@depthsection}{1}
3106 \newcommand*{\LWR@depthsubsection}{2}
3107 \newcommand*{\LWR@depthsubsubsection}{3}
3108 \newcommand*{\LWR@depthparagraph}{4}
3109 \newcommand*{\LWR@depth subparagraph}{5}
```

used by \itemize, \enumerate, \description:

```
3110 \newcommand*{\LWR@depthlist}{6}
```

used by \item:

```
3111 \newcommand*{\LWR@depthlistitem}{7}
```

31.2 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.

```
3112 \newcommand*{\LWR@closeone}{}% top of the stack
```

```

3113 \newcommand*{\LWR@closetwo}{}
3114 \newcommand*{\LWR@closethree}{}
3115 \newcommand*{\LWR@closefour}{}
3116 \newcommand*{\LWR@closefive}{}
3117 \newcommand*{\LWR@closesix}{}
3118 \newcommand*{\LWR@closeseven}{}
3119 \newcommand*{\LWR@closeeight}{}
3120 \newcommand*{\LWR@closenine}{}
3121 \newcommand*{\LWR@closeten}{}
3122 \newcommand*{\LWR@closeeleven}{}
3123 \newcommand*{\LWR@closetwelve}{}

```

31.3 Closing depths

A stack to record the depth of each level:

 Note that nested L^AT_EX structures may push depths which are non-sequential.

Ex:

```

\begin{itemize}
    \item{A}
        \begin{description}
            \item{B}
        \end{description}
\end{itemize}

```

```

3124 \newcommand*{\LWR@closedepthonenone}{\LWR@depthnone}%
3125 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
3126 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
3127 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
3128 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
3129 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
3130 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
3131 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
3132 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
3133 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
3134 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3135 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}

```

31.4 Pushing and popping the stack

```
\pushclose {<action>} {<depth>}
```

Pushes one return action and its L^AT_EX depth onto the stacks.

```

3136 \NewDocumentCommand{\pushclose}{m m}
3137 {
3138 \let\LWR@closetwelve\LWR@closeeleven
3139 \let\LWR@closeeleven\LWR@closeten
3140 \let\LWR@closeten\LWR@closenine
3141 \let\LWR@closenine\LWR@closeeight
3142 \let\LWR@closeeight\LWR@closeseven
3143 \let\LWR@closeseven\LWR@closesix
3144 \let\LWR@closesix\LWR@closefive
3145 \let\LWR@closefive\LWR@closefour
3146 \let\LWR@closefour\LWR@closethree
3147 \let\LWR@closethree\LWR@closetwo
3148 \let\LWR@closetwo\LWR@closeone
3149 \let\LWR@closeone#1
3150 \let\LWR@closedepthtwelve\LWR@closedeptheleven
3151 \let\LWR@closedepthelevel\LWR@closedepthten
3152 \let\LWR@closedepthten\LWR@closedepthnine
3153 \let\LWR@closedepthnine\LWR@closedeptheight
3154 \let\LWR@closedeptheight\LWR@closedepthseven
3155 \let\LWR@closedepthseven\LWR@closedepthsix
3156 \let\LWR@closedepthsix\LWR@closedepthfive
3157 \let\LWR@closedepthfive\LWR@closedepthfour
3158 \let\LWR@closedepthfour\LWR@closedepththree
3159 \let\LWR@closedepththree\LWR@closedepthtwo
3160 \let\LWR@closedepthtwo\LWR@closedepthonne
3161 \let\LWR@closedepthonne#2
3162 }
```

\popclose Pops one action and its depth off the stacks.

```

3163 \newcommand*{\popclose}
3164 {
3165 \let\LWR@closeone\LWR@closetwo
3166 \let\LWR@closetwo\LWR@closethree
3167 \let\LWR@closethree\LWR@closefour
3168 \let\LWR@closefour\LWR@closefive
3169 \let\LWR@closefive\LWR@closesix
3170 \let\LWR@closesix\LWR@closeseven
3171 \let\LWR@closeseven\LWR@closeeight
3172 \let\LWR@closeeight\LWR@closenine
3173 \let\LWR@closenine\LWR@closeten
3174 \let\LWR@closeten\LWR@closeeleven
3175 \let\LWR@closeeleven\LWR@closetwelve
3176 \let\LWR@closedepthonne\LWR@closedepthtwo
3177 \let\LWR@closedepthtwo\LWR@closedepththree
3178 \let\LWR@closedepththree\LWR@closedepthfour
3179 \let\LWR@closedepthfour\LWR@closedepthfive
```

```

3180 \let\LWR@closedepthfive\LWR@closedepthsix
3181 \let\LWR@closedepthsix\LWR@closedepthseven
3182 \let\LWR@closedepthseven\LWR@closedeptheight
3183 \let\LWR@closedeptheight\LWR@closedepthnine
3184 \let\LWR@closedepthnine\LWR@closedepthten
3185 \let\LWR@closedepthten\LWR@closedeptheleven
3186 \let\LWR@closedeptheleven\LWR@closedepthtwelve
3187 }

3188 \end{warpHTML}

```

32 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: 3189 `\begin{warpHTML}`

```

\LWR@setexparray {<name>} {<index>} {<contents>}

3190 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
3191 \ifthenelse{\isempty{#3}}{%
3192 {\csdef{#1#2}{}}
3193 {\expandafter\edef\csname #1#2\endcsname{\expandonce{#3}}}}
3194 }

\LWR@getexparray {<name>} {<index>}

3195 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}{}}

3196 \end{warpHTML}

```

33 HTML entities

for HTML output: 3197 `\begin{warpHTML}`

HTML entities and HTML Unicode entities:

```

3198 \let\LWR@origampersand\&

\HTMLentity {<entitytag>}

3199 \newcommand*{\HTMLentity}[1]{%
3200 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3201 \begingroup%
3202 \LWR@FBcancel%
3203 \LWR@origampersand#1;%
3204 \endgroup%
3205 % \LWR@traceinfo{HTMLentity done}%
3206 }

\HTMLunicode {<hex_unicode>}

3207 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}%

\&

3208 \renewcommand*{\&}{\HTMLentity{amp}}


\textless
\textgreater

3209 \let\LWR@origtextless\textless
3210 \renewcommand*{\textless}{\HTMLentity{lt}}
3211
3212 \let\LWR@origtextgreater\textgreater
3213 \renewcommand*{\textgreater}{\HTMLentity{gt}}


3214 \end{warpHTML}

```

34 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: 3215 `\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.

```
3216 \providecommand*{\BaseJobname}{\jobname}
```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 5.7.

```
3217 \providecommand*{\HTMLFilename}{}{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 5.7.

```
3218 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

`\SetHTMLFileName` {*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.

```
3219 \newcommand*{\SetHTMLFileName}[1]{%
3220 \setcounter{LWR@htmlfilename}{#1}%
3221 }
```

`Bool FileSectionNames` Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
3222 \newbool{FileSectionNames}
3223 \booltrue{FileSectionNames}
```

```
3224 \end{warpall}
```

for HTML output: 3225 `\begin{warpHTML}`

`Ctr LWR@htmlfilename` Records the number of each HTML file as it is being created. Number 0 is the home page.

```
3226 \newcounter{LWR@htmlfilename}
3227 \setcounter{LWR@htmlfilename}{0}
```

`\LWR@htmlsectionfilename` {*htmlfilename or name*}

Prints the filename for a given section: \HTMLFilename{}filenumber/name.html

```
3228 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3229 \LWR@traceinfo{\LWR@htmlsectionfilename A}%
```

Section 0 or empty is given the home filename. The filename must be detokenized for underscores.

```
3230 \LWR@traceinfo{about to assign temp}%
3231 \edef\LWR@tempone{\#1}%
3232 \LWR@traceinfo{about to compare with ??}%
3233 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3234 {\LWR@traceinfo{found ??}}{%
3235 {\LWR@traceinfo{not found ??}}{%
3236 \LWR@traceinfo{about to compare with zero or empty}%
3237 \ifthenelse{%
3238     \equal{\LWR@tempone}{0}}{%
3239     \OR \equal{\LWR@tempone}{}}{%
3240     \OR \equal{\LWR@tempone}{??}}{%
3241 }%
3242 {%
3243     \LWR@traceinfo{\LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3244     \HomeHTMLFilename.html%
3245 }%
```

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:

```
3246 {%
3247     \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
3248     \ifthenelse{%
3249         \equal{\HTMLFilename}{} \AND
3250         \equal{\LWR@tempone}{Index} \OR
3251         \equal{\LWR@tempone}{index}%
3252     }%
3253     {%
3254         \LWR@traceinfo{prefixing the index name with an underscore.}%
3255         \_{\#1}.html%
3256     }%
```

Otherwise, create a filename with the chosen prefix:

```
3257     {\HTMLFilename\#1.html}%
3258 }%
3259 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
3260 }
```

\LWR@htmlrefsectionfilename {\langle label \rangle}

Prints the filename for the given label

```
3261 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3262 \LWR@traceinfo{\LWR@htmlrefsectionfilename A: !#1!}%
3263 \LWR@htmlsectionfilename{\LWR@htmlfileref{\#1}}%
3264 \LWR@traceinfo{\LWR@htmlrefsectionfilename B}%
3265 }

3266 \end{warpHTML}
```

35 Homepage link

for HTML output: 3267 \begin{warpHTML}

\LinkHome \LinkHome may be used wherever you wish to place a link back to the homepage.
The filename must be detokenized for underscores.

```
3268 \newcommand*{\LinkHome}{%
3269 \LWR@subhyperrefclass{%
3270 \HomeHTMLFilename.html}%
3271 {Home}{linkhome}}%
3272 }
```

\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.

```
3273 \newcommand*{\LWR@topnavigation}{%
3274 \LWR@htmlelementclassline{nav}{topnavigation}{\LinkHome}%
3275 }
```

\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.

```
3276 \newcommand*{\LWR@botnavigation}{%
3277 \LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}%
3278 }
```

```
3279 \end{warpHTML}
```

36 \PrintStack diagnostic tool

 Diagnostics tool: Prints the L^AT_EX nesting depth values for the stack levels. Must have \LWR@startpars active while printing the stack, so \PrintStack may be called from anywhere in the normal text flow.

for HTML output: 3280 \begin{warpHTML}

\PrintStack Prints the closeddepth stack.

```
3281 \newcommand*\PrintStack{%
3282 \LWR@startpars
3283 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3284 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3285 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3286 \LWR@closedepthten{} \LWR@closedeptheleven{} \LWR@closedepthtwelve{}
3287 }
```

3288 \end{warpHTML}

37 Closing stack levels

for HTML output: 3289 \begin{warpHTML}

Close one nested level:

```
3290 \newcommand*\LWR@closeoneprevious}{%
3291
3292 \LWR@closeone{}
3293
3294 \popclose{}
3295 }
```

\LWR@closeprevious {<depth>} Close everything up to the given depth:

3296 \newcommand*\LWR@closeprevious}[1]{

Close any pending paragraph:

3297 \LWR@stoppars

Close anything nested deeper than the desired depth:

```
3298 \whiledo{\not\(\LWR@closedepthone<\#1\)}{\LWR@closeoneprevious}
3299 }

3300 \end{warpHTML}
```

38 PDF pages and styles

for HTML output: 3301 \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.

```
3302 \newcommand{\LWR@forcenewpage}{%
3303 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
3304 }
```

\pagestyle, etc. are nullified for HTML output.

\pagestyle {\<style>}

```
3305 \renewcommand*{\pagestyle}[1]{}
```

\thispagestyle {\<style>}

```
3306 \renewcommand*{\thispagestyle}[1]{}
```

\markboth {\<left>} {\<right>}

```
3307 \renewcommand*{\markboth}[2]{}
```

\markright {\<right>}

```
3308 \renewcommand*{\markright}[1]{}
```

\raggedbottom

```
3309 \renewcommand*{\raggedbottom}{}{}
```

```
\flushbottom
3310 \renewcommand*{\flushbottom}{}}

\sloppy
3311 \renewcommand*{\sloppy}{}}

\fussy
3312 \renewcommand*{\fussy}{}}

\pagenumbering {⟨commands⟩}
3313 \renewcommand*{\pagenumbering}[1]{}}

3314 \end{warpHTML}
```

39 HTML tags, spans, divs, elements

for HTML output: 3315 \begin{warpHTML}

39.1 Mapping L^AT_EX Sections to HTML Sections

```
3316 \newcommand*{\LWR@tagpart}{h2}
3317 \newcommand*{\LWR@tagpartend}{/h2}
3318 \newcommand*{\LWR@tagchapter}{h3}
3319 \newcommand*{\LWR@tagchapterend}{/h3}
3320 \newcommand*{\LWR@tagsection}{h4}
3321 \newcommand*{\LWR@tagsectionend}{/h4}
3322 \newcommand*{\LWR@tagsubsection}{h5}
3323 \newcommand*{\LWR@tagsubsectionend}{/h5}
3324 \newcommand*{\LWR@tagsubsubsection}{h6}
3325 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3326 \newcommand*{\LWR@tagparagraph}{span class="paragraph"{}}
3327 \newcommand*{\LWR@tagparagraphend}{/span}
3328 \newcommand*{\LWR@tag subparagraph}{span class="subparagraph"{}}
3329 \newcommand*{\LWR@tag subparagraphend}{/span}
3330
3331 \newcommand*{\LWR@tag regular paragraph}{p}
```

39.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for pdflatex and xelatex.

(Based on original code by DANIEL FLIPO.)

```
3332 \providecommand*\{LWR@FBcancel\}{}  
3333  
3334 \AtBeginDocument{  
3335 \@ifundefined{frenchbsetup}{  
3336 }{  
3337 {  
3338   \frenchbsetup{FrenchFootnotes=false}  
3339   % OR: redefine \insertfootnotemarkFB?  
3340   \LetLtxMacro\{LWR@FBcancel\}{NoAutoSpacing}  
3341   \renewcommand*\{FBcolonspace\}{%  
3342     \begingroup%  
3343     \LWR@FBcancel%  
3344     \LWR@origampersand{}nbsp;%  
3345     \endgroup%  
3346   }  
3347   \renewcommand*\{FBthinspace\}{%  
3348     \begingroup%  
3349     \LWR@FBcancel%  
3350     \LWR@origampersand\x202f;% \,  
3351     \endgroup%  
3352   }  
3353   \renewcommand*\{FBguillspace\}{%  
3354     \begingroup%  
3355     \LWR@FBcancel%  
3356     \LWR@origampersand{}nbsp;% ~, for \og xyz \fg{}  
3357     \endgroup%  
3358 }  
3359 \DeclareDocumentCommand\{FBmedkern\}{}{  
3360   \begingroup%  
3361   \LWR@FBcancel%  
3362   \LWR@origampersand\x202f;% \,  
3363   \endgroup%  
3364 }  
3365 \DeclareDocumentCommand\{FBthickkern\}{}{  
3366   \begingroup%  
3367   \LWR@FBcancel%  
3368   \LWR@origampersand{}nbsp;% ~  
3369   \endgroup%  
3370 }  
3371 \renewcommand*\{\~{}\{HTMLentity{nbsp}\}}% was overwritten by babel-french  
3372 \ifFBunicode  
3373 \else
```

```

3374     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}
3375     \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}
3376 \fi
3377 }
3378 }
```

39.3 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.

```

3379 \newcommand*{\LWR@htmltagc}[1]{%
3380 {%
3381 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3382 \begingroup%
3383 \LWR@FBcancel%
3384 \ifmmode\else\protect\LWR@origttfamily\fi%
3385 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3386 \endgroup%
3387 }%
3388 }
```

Env LWR@nestspan Disable minipage, \parbox, and HTML <div>s inside a .

⚠ \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.

⚠

```

3389 \newcommand*{\LWR@nestspanitem}{%
3390 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
3391 \LWR@origitem%
3392 }%
3393
3394 \newenvironment*{\LWR@nestspan}{%
3395 {%
3396 \LWR@traceinfo{\LWR@nestspan}%
3397 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
3398 {%
3399 \LWR@traceinfo{\LWR@nestspan: inside a lateximage}%
3400 }%
3401 {%
3402 \LWR@traceinfo{\LWR@nestspan: NOT inside a lateximage}%
3403 \addtocounter{\LWR@spandepth}{1}%

```

```

3404 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{\begin{minipage}[#1]{#2}}{\end{minipage}}
3405 \RenewDocumentEnvironment{BlockClass}{o m}{\begin{BlockClass}[#1]}{\end{BlockClass}}
3406 \renewcommand{\BlockClassSingle}[2]{\begin{BlockClass}[#1]\end{BlockClass}}
3407 \renewcommand{\LWR@forcenewpage}{\begin{LWR@nestspan}\end{LWR@nestspan}}
3408 \renewcommand{\LWR@itemizestart}{\begin{LWR@nestspan}\begin{itemize}}
3409 \let\item\begin{LWR@nestspan}\item\end{LWR@nestspan}
3410 \end{itemize}\end{LWR@nestspan}}
3411 \renewcommand{\LWR@itemizeend}{\end{itemize}\end{LWR@nestspan}}
3412 \renewcommand{\LWR@enumeratestart}{\begin{LWR@nestspan}\begin{enumerate}}
3413 \let\item\begin{LWR@nestspan}\item\end{LWR@nestspan}
3414 \end{enumerate}\end{LWR@nestspan}}
3415 \renewcommand{\LWR@enumerateend}{\end{enumerate}\end{LWR@nestspan}}
3416 \renewcommand{\LWR@descriptionstart}{\begin{LWR@nestspan}\begin{description}}
3417 \let\item\begin{LWR@nestspan}\item\end{LWR@nestspan}
3418 \end{description}\end{LWR@nestspan}}
3419 \renewcommand{\LWR@descriptionend}{\end{description}\end{LWR@nestspan}}
3420 }% not in a lateximage
3421 }%
3422 {%
3423 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}>0}{%
3424 }%
3425 {\addtocounter{LWR@spandepth}{-1}}%
3426 \LWR@traceinfo{LWR@nestspan: done}%
3427 }
3428
3429 \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.

```

3430 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3431 \LWR@ensuredoingapar%
3432 \LWR@htmntagc{#1}%
3433 \begin{LWR@nestspan}%
3434 #2%
3435 \LWR@htmntagc{/#1}%
3436 \end{LWR@nestspan}%
3437 }

```

\LWR@htmlspanclass [*style*] {*class*} {*text*}

```

3438 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3439 \LWR@ensuredoingapar%
3440 \LWR@subhtmlelementclass{span}[#1]{#2}%
3441 \begin{LWR@nestspan}%

```

```

3442 #3%
3443 \LWR@htmtagc{/span}%
3444 \end{LWR@nestspan}%
3445 }

\LWR@htmtag {<tag>}

Print an HTML tag: <tag>

3446 \newcommand*{\LWR@htmtagb}[1]{%
3447 \LWR@htmtagc{#1}%
3448 \endgroup%
3449 }
3450
3451 \newcommand*{\LWR@htmtag}{%
3452 \begingroup\catcode`\_=12
3453 \LWR@htmtagb%
3454 }

```

39.4 Block tags and comments

In the following, `\origttfamily` breaks ligatures, which may not be used for HTML codes:

```

\LWR@htmlopencomment
\LWR@htmclosecomment
3455 \newcommand*{\LWR@htmlopencomment}{%
3456 {%
3457 % \LWR@traceinfo{\LWR@htmlopencomment}%
3458 \begingroup%
3459 \LWR@FBcancel%
3460 \ifmmode\else\protect\LWR@origttfamily\fi%
3461 \LWR@origtextless{}!{-}{-}%
3462 \endgroup%
3463 }%
3464 }
3465
3466 \newcommand*{\LWR@htmclosecomment}{%
3467 {%
3468 % \LWR@traceinfo{\LWR@htmclosecomment}%
3469 \begingroup%
3470 \LWR@FBcancel%
3471 \ifmmode\else\protect\LWR@origttfamily\fi%
3472 {-}{-}\LWR@origtextgreater{}%
3473 \endgroup%
3474 }%

```

```
3475 }
```

```
\LWR@htmlcomment  {\langle comment \rangle}
```

```
3476 \newcommand{\LWR@htmlcomment}[1]{%
3477 \LWR@htmlopencomment{}%
3478 {%
3479 \LWR@origttfamily% break ligatures
3480 #1%
3481 }%
3482 \LWR@htmlclosecomment{}}
```

```
\LWR@htmlblockcomment  {\langle comment \rangle}
```

```
3483 \newcommand{\LWR@htmlblockcommentb}[1]%
3484 {\LWR@stoppars\LWR@htmlcomment{\#1}\LWR@startpars\endgroup}%
3485
3486 \newcommand{\LWR@htmlblockcomment}%
3487 {%
3488 \begingroup\catcode`\_=12%
3489 \LWR@htmlblockcommentb%
3490 }
```

```
\LWR@htmlblocktag  {\langle tag \rangle} print a stand-alone HTML tag
```

```
3491 \newcommand*{\LWR@htmlblocktag}[1]{%
3492 \LWR@stoppars%
3493 \LWR@htmntag{\#1}%
3494 \LWR@startpars%
3495 }
```

39.5 Div class and element class

```
\LWR@subhtmlelementclass  {\langle element \rangle} [\langle style \rangle] {\langle class \rangle}
```

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
3496 \NewDocumentCommand{\LWR@subhtmlelementclass}{m o m}{%
3497 \IfValueTF{\#2}{%
3498 {%
3499     \ifthenelse{\equal{\#2}{}}{%
3500         {\LWR@htmntag{\#1 class="#3" }}%
3501     }%
3502     empty option
3503 }}
```

```

3501     {\LWR@htmntag{\#1 class="#3" style="#2" }}% non-empty option
3502 }% option
3503 {\LWR@htmntag{\#1 class="#3" }}% no option
3504 }

\LWR@htmlelementclass {<element>} {<class>} [<style>]

3505 \NewDocumentCommand{\LWR@htmlelementclass}{m o m}{%
3506 \LWR@stoppars%
3507 \LWR@subhtmlelementclass{\#1}[\#2]{\#3}%
3508 \LWR@startpars%
3509 }

\LWR@htmlelementclassend {<element>} {<class>}

3510 \newcommand*{\LWR@htmlelementclassend}[2]{%
3511 \LWR@stoppars%
3512 \LWR@htmntag{/#1}%
3513 \ifbool{HTMLDebugComments}{%
3514     \LWR@htmlcomment{End of #1 ` '#2' }%
3515 }{}%
3516 \LWR@startpars%
3517 }

\LWR@htmldivclass [<style>] {<class>}

3518 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3519 \LWR@htmlelementclass{div}[\#1]{\#2}%
3520 }

\LWR@htmldivclassend {<class>}

3521 \newcommand*{\LWR@htmldivclassend}[1]{%
3522 \LWR@htmlelementclassend{div}{\#1}%
3523 }

```

39.6 Single-line elements

A single-line element, without a paragraph tag for the line of text:

```

\LWR@htmlelementclassline {<element>} [<style>] {<class>} {<text>}

3524 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%

```

```

3525 \LWR@stoppars
3526 \LWR@subhtmlelementclass{#1}{#2}{#3}%
3527 #4%
3528 \LWR@htmntag{/#1}
3529 \LWR@startpars
3530 }

```

39.7 HTML5 semantic elements

```

\LWR@htmlelement {<element>}

3531 \newcommand*{\LWR@htmlelement}[1]{%
3532 \LWR@htmlblocktag{#1}
3533 }

\LWR@htmlelementend {<element>}

3534 \newcommand*{\LWR@htmlelementend}[1]{%
3535 \LWR@stoppars
3536 \LWR@htmntag{/#1}
3537 \LWR@startpars
3538 }
3539
3540 \end{warpHTML}

```

39.8 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.

For other direct-formatting commands, see section [72](#).

Env **BlockClass** [*style*] {*class*} High-level interface for <div> classes.

Ex: \begin{BlockClass}{class} text \end{BlockClass}

for HTML output: 3541 \begin{warpHTML}
3542 \NewDocumentEnvironment{BlockClass}{o m}%
3543 {
3544 \LWR@htmldivclass[#1]{#2}
3545 }
3546 {
3547 \LWR@htmldivclassend{#2}

```

3548 }
3549 \end{warpHTML}

for PRINT output: 3550 \begin{warpprint}
3551 \NewDocumentEnvironment{BlockClass}{o m}{}{%
3552 \end{warpprint}

\BlockClassSingle {<class>} {<text>} A single-line <div>, without a paragraph tag for the line of text.

for HTML output: 3553 \begin{warpHTML}
3554 \newcommand{\BlockClassSingle}[2]{%
3555 \LWR@htmlelementclassline{div}{#1}{#2}%
3556 }
3557 \end{warpHTML}

for PRINT output: 3558 \begin{warpprint}
3559 \newcommand{\BlockClassSingle}[2]{#2}
3560 \end{warpprint}

\InlineClass {[style]} {<class>} {<text>} High-level interface for inline span classes.

for HTML output: 3561 \begin{warpHTML}
3562 \NewDocumentCommand{\InlineClass}{o m +m} {%
3563 \LWR@htmlspanclass[#1]{#2}{#3}%
3564 }
3565 \end{warpHTML}

for PRINT output: 3566 \begin{warpprint}
3567 \NewDocumentCommand{\InlineClass}{o m +m} {#3}%
3568 \end{warpprint}

```

39.9 Closing HTML tags

```

for HTML output: 3569 \begin{warpHTML}

Sections H1, H2, etc. do not need a closing HTML tag, but we add a comment for
readability:

3570 \newcommand*{\LWR@printclosepart}%
3571   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3572 \newcommand*{\LWR@printclosechapter}%
3573   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3574 \newcommand*{\LWR@printclosesection}%
3575   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3576 \newcommand*{\LWR@printclosesubsection}%
3577   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}

```

```

3578 \newcommand*{\LWR@printclosesubsubsection}{}
3579   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}{}}
3580 \newcommand*{\LWR@printcloseparagraph}{}
3581   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}{}}
3582 \newcommand*{\LWR@printclosesubparagraph}{}
3583   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}{}}

```

Lists require closing HTML tags:

```

3584 \newcommand*{\LWR@printcloselistitem}{}
3585   {\LWR@htmllisttag{/li}}
3586 \newcommand*{\LWR@printclosedescitem}{}
3587   {\LWR@htmllisttag{/dd}}
3588 \newcommand*{\LWR@printcloseitemize}{}
3589   {\LWR@htmllisttag{/ul}}
3590 \newcommand*{\LWR@printcloseenumerate}{}
3591   {\LWR@htmllisttag{/ol}}
3592 \newcommand*{\LWR@printclosedescription}{}
3593   {\LWR@htmllisttag{/dl}}

```

```
3594 \end{warpHTML}
```

40 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 41 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a L^AT_EX paragraph is being used (`\LWR@doingapar`). `\LWR@lateximagedepth` is used to prevent nesting tags inside a `\textrm`. `\LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output: 3595 `\begin{warpHTML}`

Ctr `\LWR@spandepth` Do not create paragraph tags inside of an HTML span.

```

3596 \newcounter{\LWR@spandepth}
3597 \setcounter{\LWR@spandepth}{0}

```

Bool `\LWR@doingstartpars` Tells whether paragraphs may be generated.

```

3598 \newbool{\LWR@doingstartpars}
3599 \boolfalse{\LWR@doingstartpars}

```

Bool LWR@doingapar Tells whether have actually generated and are currently processing paragraph text.

```
3600 \newbool{LWR@doingapar}
3601 \global\boolearn{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:

```
3602 \newcommand*\LWR@ensuredoingapar{%
3603 \ifbool{LWR@doingstartpars}{%
3604 {\global\boolearn{LWR@doingapar}}{%
3605 {}{%
3606 }}
```

\LWR@openparagraph

```
3607 \newcommand*\LWR@openparagraph{%
3608 {}%
```

See if paragraph handling is enabled:

```
3609 \ifbool{LWR@doingstartpars}{%
3610 {}% handling pars}
```

See if have already started a `lateximage` or a ``. If so, do not generate nested paragraph tags.

```
3611 \ifthenelse{%
3612   \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3613   \cnttest{\value{LWR@spandepth}}{>}{0}%
3614 }% 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 ``.

```
3615 {}% no nested par tags
```

Else: No `lateximage` or `` has been started yet, so it's OK to generate paragraph tags.

```
3616 {}% yes nest par tags
3617 \LWR@htmlltagc{\LWR@tagregularparagraph}%
```

Manually indent item list labels to avoid left margin intrustion:

See if are nested inside an item list:

```
3618     \ifnumcomp{\@listdepth}{>}{0}%
3619     {%
```

If so, leave some horizontal room in the L^AT_EX PDF output for list labels:

```
3620     \LWR@orighspace{1in}%
3621     }%
3622     {}%
```

Now have started a paragraph.

```
3623     \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.)

```
3624     \let\par\LWR@closeparagraph%
3625     }% end of yes nest par tags
3626 }% end of handling pars
3627 {}% not handling pars
3628 }
```

\LWR@closeparagraph

```
3629 \newcommand*{\LWR@closeparagraph}%
3630 {%
```

See if paragraph handling is enabled:

```
3631 \ifbool{LWR@doingapar}{}%
```

If currently in paragraph mode:

```
3632 {}% handling pars
```

See if already started a lateximage or a :

```
3633     \ifthenelse{%
3634         \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3635         \cnttest{\value{LWR@spandepth}}{>}{0}%
3636     }%
```

Do nothing if already started a lateximage or a , but add a parbreak if in a span but not a lateximage.

```
3637 {}% no nested par tags
```

```

3638      \ifthenelse{%
3639          \cnttest{\value{LWR@spandepth}}{>}{0}%
3640          \AND%
3641          \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3642      }%
3643      {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}}%
3644      {}%
3645  }% no nested par tags

```

If have not already started a `lateximage` or a ``:

```
3646  {}% yes nest par tags
```

Print a closing tag and some extra vertical space:

```

3647  \unskip%
3648  \LWR@htmltagc{/LWR@tagregularparagraph}%
3649  \LWR@newline%

```

No longer doing a paragraph:

```

3650  \global\boolfalse{LWR@doingapar}%
3651 % Disable the special \env{minipage} \& \cs{hspace} interaction
3652 % until a new minipage is found:
3653 %  \begin{macrocode}
3654  \global\boolfalse{LWR@minipagethispar}%
3655  }% end of yes nest par tags
3656 }% end of handling pars

```

Add a parbreak if in a span, but not in a table outside a row:

```

3657 {}% not handling pars
3658  \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0} }{%
3659  {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}}%
3660  {}%
3661 }% not handling pars

```

Finish with regular paragraph processing

```

3662 \LWR@origpar%
3663 }

```

```
3664 \end{warpHTML}
```

41 Paragraph start/stop handling

These commands allow/disallow the generation of HTML paragraph tags.

Section 40 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: 3665 `\begin{warpHTML}`

`\LWR@startpars` Begin handling HTML paragraphs. This allows an HTML paragraph to start, but one has not yet begun.

```
3666 \newcommand*{\LWR@startpars}{%
3667 {%
```

Ignore if inside a span:

```
3668 \ifthenelse{\cnttest{\value{\LWR@spandepth}}{>}{0}}{%
3669 {}%
3670 {}%
```

See if currently handling HTML paragraphs:

```
3671 \ifbool{\LWR@doingstartpars}{%
```

If already in paragraph mode, do nothing.

```
3672 {}%
```

If not currently in paragraph mode:

```
3673 {}%
```

At the start of each paragraph, generate an opening tag:

```
3674 \PushPreHook{par}{\LWR@openparagraph}{%
```

At the end of each paragraph, generate closing tag and do regular /par actions:

```
3675 \let\par\LWR@closeparagraph
3676 }% an intentionally blank line
```

Are now handling paragraphs, but have not yet actually started one:

```
3678     \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
3679     \global\boolfalse{LWR@doingapar}%
3680 }% nestspan
3681 }
```

\LWR@stoppars Stop handling HTML paragraphs. Any currently open HTML paragraph is closed, and no more will be opened.

```
3682 \newcommand*\LWR@stoppars{%
3683 {%
```

Ignore if inside a span:

```
3684 \ifthenelse{\cnttest{\value{LWR@spandepth}}>{0}}{%
3685 {}%
3686 {}%
```

See if currently handling HTML paragraphs:

```
3687 \ifbool{LWR@doingapar}{%
```

if currently in an HTML paragraph:

```
3688 {}%
```

Print a closing tag:

```
3689 \unskip%
3690 \LWR@htmlltagc{/LWR@tagregularparagraph}%
3691 \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
3692 \global\boolfalse{LWR@doingapar}%
```

Disable the special `\minipage` & `\hspace` interaction until a new minipage is found:

```
3693 \global\boolfalse{LWR@minipagethispar}
3694 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3696 {}%
```

See if currently allowing HTML paragraphs:

```
3697 \ifbool{LWR@doingstartpars}{%
```

If so: clear the par hook to no longer catch paragraphs:

```
3698 {\ClearPreHook{par}}{%
```

Else: do nothing

```
3699 {}{}
```

no longer in paragraph mode

```
3700 \global\setbool{LWR@doingstartpars}{false}{%
```

no <p> tag to undo:

```
3701 \global\boolfalse{LWR@doingapar}{%
3702 }{} nestspan
3703 }
```

```
3704 \end{warpHTML}
```

42 Page headers and footers

for HTML & PRINT: 3705 \begin{warpall}

In the following, catcode is manually changes back and forth without groups, since new macros are being defined which must not be contained within the groups.

```
3706 \newcommand{\LWR@firstpagetop}{} % for the home page alone
3707 \newcommand{\LWR@pagetop}{} % for all other pages
3708 \newcommand{\LWR@pagebottom}{}%
3709
3710 \newcommand{\LWR@setfirstpagetopb}[1]{%
3711 \renewcommand{\LWR@firstpagetop}{#1}
3712 \catcode`\_=8
3713 }
```

```
\HTMLFirstPageTop {\langle text and logos\rangle}
```

```
3714 \newcommand{\HTMLFirstPageTop}{%
3715 \catcode`\_=12
```

```

3716 \LWR@setfirstpagetopb
3717 }

3718 \newcommand{\LWR@setpagetopb}[1]{%
3719 \renewcommand{\LWR@pagetop}{#1}
3720 \catcode`\_=8
3721 }

\HTMLPageTop {<text and logos>}

3722 \newcommand{\HTMLPageTop}{%
3723 \catcode`\_=12
3724 \LWR@setpagetopb
3725 }

3726 \newcommand{\LWR@setpagebottomb}[1]{%
3727 \renewcommand{\LWR@pagebottom}{#1}
3728 \catcode`\_=8
3729 }

\HTMLPageBottom {<text and logos>}

3730 \newcommand{\HTMLPageBottom}{%
3731 \catcode`\_=12
3732 \LWR@setpagebottomb
3733 }

3734 \end{warpall}

```

43 CSS

for HTML output: 3735 \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.

```
3736 \newcommand*{\LWR@currentcss}{lwarp.css}
```

\CSSFilename {<new-css-filename.css>} Assigns the css file to be used by the following HTML pages.

```
3737 \newcommand*{\LWR@newcssb}[1]{%
```

```

3738 \renewcommand*\LWR@currentcss}{#1}
3739 \catcode`\_=8
3740 }
3741
3742 \newcommand*\CSSfilename{
3743 \catcode`\_=12
3744 \LWR@newcssb
3745 }
3746 \end{warpHTML}

for PRINT output: 3747 \begin{warpprint}
3748 \newcommand*\CSSfilename[1]{}
3749 \end{warpprint}

```

44 HTML meta description and author

for HTML & PRINT: 3750 \begin{warpall}

\HTMLAuthor {<authorname>} The author to place into an HTML meta tag.

```

3751 \providetcommand{\theauthor}{}
3752 \newcommand{\theHTMLAuthor}{\theauthor}
3753
3754 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
3755 \end{warpall}

```

for HTML & PRINT: 3756 \begin{warpall}

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.

```

3757 \newcommand{\LWR@currentHTMLDescription}{}
3758
3759 \newcommand{\HTMLDescription}[1]{%
3760 \renewcommand{\LWR@currentHTMLDescription}{#1}
3761 }
3762
3763 \end{warpall}

```

45 Footnotes

lwarf uses native L^AT_EX footnote code, although with its own \box to avoid the L^AT_EX output routine. The usual functions work as-is.

Several kinds of footnotes are used: in a regular page, in a minipage, or as thanks in the titlepage. Each of these is handled differently.

45.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the L^AT_EX 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 45.4 for the implementation.

45.2 Minipage footnotes

See section 45.5 for how minipage footnotes are gathered. See section 71.3 for how minipage footnotes are placed into the document.

45.3 Titlepage thanks

See section 52.6 for titlepage footnotes.

45.4 Regular page footnote implementation

for HTML output: 3764 \begin{warpHTML}

Patch L^AT_EX footnotes to use a new \box for lwarf footnotes.

3765 \newbox{\LWR@footnotes}

Much of the following has unneeded print-mode formatting removed.

\@makefntext {<text>}

3766 \long\def\@makefntext#1{\@thefnmark #1}

```
\@makefnmark
3767 \def\@makefnmark{\hbox{\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\WR@footnotetext
```

`\WR@footnotetext`

```
{<text>}
3768 \long\def\WR@footnotetext#1{%
3769 \global\setbox\WR@footnotes=\vbox{%
```

Add to any current footnotes:

```
3770 \unvbox\WR@footnotes%
```

Remember the footnote number for `\ref`:

```
3771 \protected@edef\@currentlabel{%
3772     \csname p@footnote\endcsname\@thefnmark%
3773 }% @currentlabel
```

Open a group:

```
3774 \color@begingroup%
```

Use HTML superscripts even inside a `lateximage`:

```
3775 \renewcommand{\textsuperscript}[1]{\WR@htmlspan{sup}{##1}}%
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
3776 \ifthenelse{%
3777     \boolean{\WR@doingstartpars} \AND%
3778     \cnttest{\value{\WR@lateximagedepth}}{=}{0}%
3779 }%
3780 {}%
3781 {\WR@htmltagc{\WR@tagregularparagraph}}%
```

Append the footnote to the list:

```
3782     \@makefntext{\#1}%
```

Closing paragraph tag:

```
3783     \ifthenelse{%
3784         \boolean{LWR@doingstartpars} \AND%
3785         \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3786     }{%
3787     {\par}%
3788     {%
3789         \LWR@htmlltagc{/ \LWR@tagregularparagraph}%
3790         \LWR@orignewline%
3791     }%
}
```

Close the group:

```
3792     \color@endgroup%
3793 }% vbox
3794 }%
```

```
\@footnotetext {<text>}
```

```
3795 \let\@footnotetext\LWR@footnotetext
```

45.5 Minipage footnote implementation

```
\@mpfootnotetext {<text>}
```

```
3796 \long\def\@mpfootnotetext#1{%
3797 \global\setbox\@mpfootins\vbox{%
3798     \unvbox\@mpfootins
3799     \reset@font\footnotesize
3800     \hsize\columnwidth
3801     \parboxrestore
3802     \protected@edef\@currentlabel
3803     {\csname p@mpfootnote\endcsname\@thefnmark}%
3804     \color@begingroup
```

Use paragraph tags if in a tabular data cell or a `lateximage`:

```
3805     \ifthenelse{%
3806         \boolean{LWR@doingstartpars} \AND%
3807         \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3808     }{%
```

```

3808      }%
3809      {}%
3810      {\LWR@htmlltagc{\LWR@tagregularparagraph}}%
3811      \@makefntext{%
3812          \ignorespaces#1%
3813      }%

```

Don't add the closing paragraph tag if are inside a `lateximage`:

```

3814      \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
3815      {}%
3816      {}%
3817      \LWR@htmlltagc{/}\LWR@tagregularparagraph}%
3818      \LWR@orignewline%
3819      }%
3820      \color@endgroup%
3821 }% vbox
3822 }

```

45.6 Printing pending footnotes

`\LWR@printpendingfootnotes` Enclose the footnotes in a class, print, then clear.

```

3823 \newcommand*{\LWR@printpendingfootnotes}{%
3824 \ifvoid\LWR@footnotes\else
3825     \LWR@forcenewpage
3826     \begin{BlockClass}{footnotes}
3827     \LWR@origmedskip
3828     \unvbox\LWR@footnotes
3829     \setbox\LWR@footnotes=\vbox{}
3830     \end{BlockClass}
3831 \fi
3832 }

```

`\LWR@epubprintpendingfootnotes` Used to print footnotes before sections only if formatting for an EPUB or word processor:

```

3833 \newcommand*{\LWR@epubprintpendingfootnotes}{%
3834 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{%
3835 {\LWR@printpendingfootnotes}%
3836 {}%
3837 }%
3838 \end{warpHTML}

```

46 Marginpars

```
for HTML output: 3839 \begin{warpHTML}

\marginpar [⟨left⟩] {⟨right⟩}

3840 \renewcommand{\marginpar}[2] []{%
3841 \LWR@htmlspanclass{marginpar}{#2}%
3842 }

3843 \end{warpHTML}
```

47 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: 3844 \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.

```
3845 \newcounter{FileDepth}
3846 \setcounter{FileDepth}{-5}
```

`Bool CombineHigherDepths` Combile higher-level sections together into one file?

```
3847 \newbool{CombineHigherDepths}
3848 \booltrue{CombineHigherDepths}

3849 \end{warpall}
```

for HTML output: 3850 \begin{warpHTML}

`\LWR@thisfilename` The currently-active filename or number.

```
3851 \newcommand*{\LWR@thisfilename}{}{}
```

\LWR@thisfilename The filename being sanitized.

```
3852 \newcommand*{\LWR@thisfilename}{}%
```

\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 \TeX commands which appear in section names and TOC captions.

```
3853 \newcommand*{\LWR@filenamenoblanks}[1]{%
3854 \begingroup
```

Locally temporarily disable direct-formatting commands, not used in filenames:

```
3855 \LWR@nullfonts
3856 \renewcommand*{\LWR@htmllagc}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
3857 \edef\LWR@thisfilename{#1}
3858 \fullexpandarg
```

Convert spaces into hyphens:

```
3859 \StrSubstitute{\LWR@thisfilename}{ }{-}[\LWR@thisfilename]
```

Convert punctuation into hyphens:

```
3860 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
3861 \StrSubstitute{\LWR@thisfilename}{'}{-}[\LWR@thisfilename]
3862 \StrSubstitute{\LWR@thisfilename}{%}
3863 {\LWR@origampersand}{-}[\LWR@thisfilename]
3864 \StrSubstitute{\LWR@thisfilename}{+}{-}[\LWR@thisfilename]
3865 \StrSubstitute{\LWR@thisfilename}{,}{-}[\LWR@thisfilename]
3866 \StrSubstitute{\LWR@thisfilename}{/}{-}[\LWR@thisfilename]
3867 \StrSubstitute{\LWR@thisfilename}{:}{-}[\LWR@thisfilename]
3868 \StrSubstitute{\LWR@thisfilename}{;}{-}[\LWR@thisfilename]
3869 \StrSubstitute{\LWR@thisfilename}{=}{-}[\LWR@thisfilename]
3870 \StrSubstitute{\LWR@thisfilename}{?}{-}[\LWR@thisfilename]
3871 \StrSubstitute{\LWR@thisfilename}{@}{-}[\LWR@thisfilename]
3872 \StrSubstitute{\LWR@thisfilename}{"}{-}[\LWR@thisfilename]
3873 \StrSubstitute{\LWR@thisfilename}{%}
3874 {\textless}{-}[\LWR@thisfilename]
3875 \StrSubstitute{\LWR@thisfilename}{%}
```

```

3876 {\textgreater}{-}[\LWR@thisnewfilename]
3877 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3878 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3879 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
3880 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
3881 \StrSubstitute{\LWR@thisnewfilename}{\|}{-}[\LWR@thisnewfilename]
3882 \StrSubstitute{\LWR@thisnewfilename}{\%}
3883 {\textbackslash}{-}[\LWR@thisnewfilename]
3884 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
3885 \StrSubstitute{\LWR@thisnewfilename}{\~{}}{-}[\LWR@thisnewfilename]
3886 %      "\~{}" for babel
3887 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3888 \StrSubstitute{\LWR@thisnewfilename}{\}]{-}[\LWR@thisnewfilename]
3889 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

3890 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3891 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3892 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3893 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3894 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3895 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3896 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3897 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3898 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3899 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
3900 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
3901 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3902 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3903 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3904 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3905 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3906 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
3907 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

3908 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3909 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3910 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3911 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3912 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3913 %      emdash
3914 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3915 %      endash
3916 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3917 \endgroup
3918 }

```

```
\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.

```
3919 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```
3920 \LWR@htmlelementclassend{section}{textbody}
3921
3922 \LWR@printpendingfootnotes
3923
```

No footer between files if EPUB:

```
3924 \ifbool{FormatEPUB}
3925 {}
3926 {
3927     \LWR@htmlelement{footer}
3928
3929     \LWR@pagebottom
3930
3931     \LWR@htmlelementend{footer}
3932 }
```

No bottom navigation if are finishing the home page or formatting for EPUB or a word-processor.

```
3933 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWordProcessor}}
3934 {}
3935 {\ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
3936 \LWR@stopars
3937 \LWR@htmlltag{/body}\LWR@orignewline
3938 \LWR@htmlltag{/html}\LWR@orignewline
3939 \LWR@orignewpage
3940
3941 \addtocounter{LWR@htmlfilename}{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.

```
3942 \ifbool{FileSectionNames}{%
3943 {\LWR@filenamenoblanks{\#1}}
3944 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmldocument}}
```

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.

```
3945 \LWR@htmlblockcomment{%
3946 |Start file|%
3947 \LWR@htmlsectionfilename{\LWR@thisfilename}|%
3948 }
```

At the top of the starting file:

```
3949 \LWR@stoppars
3950

3951 \LWR@filestart{ -- #1}%
3952 there is an EMdash in front of the #1
```

No navigation between files if formatting for an EPUB or word processor:

```
3953 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWordProcessor}}%
3954 {}
3955 {\LWR@topnavigation}
3956
```

No header if between files if formatting for an EPUB or word processor:

```
3957 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWordProcessor}}%
3958 {}
3959 {
3960     \LWR@htmlelement{header}
3961
3962     \LWR@pagetop
3963
3964     \LWR@htmlelementend{header}
3965 }
3966
```

Print title only if there is one. Skip if formatting for an EPUB or word processor:

```
3967 \ifthenelse{\boolean{FormatEPUB} \OR \boolean{FormatWordProcessor}}%
3968 {}
3969 {\ifcsvoid{\thetitle}{}{\LWR@printthetitle}}
3970
```

No sidetoc if formatting for an EPUB or word processor:

```
3971 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3972 {}
3973 {\LWR@sidetoc}
3974
```

Start of the <textbody>:

```
3975 \LWR@htmlelementclass{section}{textbody}
3976
```

Keep paragraph tags disabled for now:

```
3977 \LWR@stoppars
3978
```

Track the page numbers:

```
3979 \setcounter{LWR@latestautopage}{\value{page}}
3980 }

3981 \end{warpHTML}
```

48 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: 3982 \begin{warpHTML}

48.1 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.

```
3983 \newbool{LWR@forcinghtmlpage}
3984 \boolfalse{LWR@forcinghtmlpage}
3985
3986 \newcommand*{\ForceHTMLPage}{%
3987 \global\booltrue{LWR@forcinghtmlpage}%
3988 }
```

`\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.

```
3989 \newbool{LWR@forcinghtmltoc}
3990 \boolfalse{LWR@forcinghtmltoc}
3991
3992 \newcommand*{\ForceHTMLTOC}{%
3993 \global\booltrue{LWR@forcinghtmltoc}%
3994 }
```

```
3995 \end{warpHTML}
```

for PRINT output: 3996 `\begin{warpprint}`
 3997 `\newcommand*{\ForceHTMLPage}{}`
 3998 `\newcommand*{\ForceHTMLTOC}{}`
 3999 `\end{warpprint}`

for HTML output: 4000 `\begin{warpHTML}`

48.2 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.

```
4001 \newbool{LWR@mainmatter}
4002 \DeclareDocumentCommand{\mainmatter}{ }{%
```

```
4003 \booltrue{LWR@mainmatter}%
4004 }
```

\frontmatter Declare the front matter section of the document, using arabic numbering for the internal numbering. Does not reset the page number.

```
4005 \DeclareDocumentCommand{\frontmatter}{}{%
4006 \boolfalse{LWR@mainmatter}%
4007 }
```

\backmatter Declare the back matter section of the document. Does not reset the page number.

```
4008 \DeclareDocumentCommand{\backmatter}{}{%
4009 \boolfalse{LWR@mainmatter}%
4010 }
```

48.3 Sectioning support macros

\LWR@sectionnumber {*<section type>*}

Typeset a section number and its trailing space with CSS formatting:

```
4011 \newcommand*{\LWR@sectionnumber}[1]{%
4012 \InlineClass{sectionnumber}{#1}%
4013 }
```

autosec A tag used by the TOC and index.

\LWR@createautosec {*<section type>*}

Create an autosection tag.

```
4014 \newcommand*{\LWR@createautosec}[1]{%
4015 \LWR@htmltag{#1 id="autosec-\thepage"{} }%
4016 }
```

\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.

```
4017 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\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@stopars must be executed first.

4018 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%

Close any stacked sections up to this new one.

4019 \LWR@closeprevious{#1}%

Push a new section depth:

4020 \LWR@pushoneclose{#1}{#2}%

4021 }

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.

4022 \newcounter{LWR@prevFileDepth}

4023 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}

\@seccntformat

4024 \def\@seccntformat#1{\csname the#1\endcsname\protect\quad}

\LWR@section * [⟨TOC name⟩] {⟨name⟩} {⟨sectiontype⟩}

The common actions for the high-level sectioning commands.

4025 \DeclareDocumentCommand{\LWR@section}{m m m m}{%

4026 \LWR@traceinfo{\LWR@section |#2| |#3|}%

4027 % \ifthenelse{%

4028 % % \equal{#2}{\-NoValue-}\AND%

4029 % \isempty{#3}%

4030 % }%

4031 % {\% empty section name

4032 % \LWR@traceinfo{\LWR@section: empty section}%

4033 % }% empty section name

4034 % {\% not an empty section name

4035 \LWR@traceinfo{\LWR@section: not an empty section}%

4036 \LWR@stopars%

Cancel special `minipage` horizontal space interaction:

```
4037 \global\boolefalse{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 L^AT_EX page so that toc and index page number points to the section:

```
4038 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4039 \IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{%
4040 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
4041 \ifthenelse{%
4042   \%
4043   (\NOT\equal{#1}{\BooleanTrue})\OR%
4044   (\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart})\OR%
4045   (\boolean{LWR@forcinghtmlpage})\%
4046   \%
4047   \AND%
4048   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}\%
4049   \AND%
4050   \%
4051   \NOT\boolean{CombineHigherDepths}\OR%
4052   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}\%
4053   \%
4054   \AND%
4055   \NOT\equal{#3}{% phantomsection
4056 }%
```

If so: start a new HTML file:

```
4057 {%
4058   new file
4059   \LWR@traceinfo{LWR@section: new HTML file}%
}
```

See if there was an optional TOC name entry:

```
4059 \IfNoValueTF{#2}{%
```

If no optional entry

```
4060 {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
4061 {\LWR@newhtmlfile{#2}}%
4062 }% new file
```

Else: No new HTML file:

```
4063 { % not new file
```

Generate a new L^AT_EX page so that toc and index page number points to the section:

```
4064     \LWR@traceinfo{LWR@section: not a new HTML file}%
4065     \LWR@orignewpage%
4066
4067 }% not new file
```

Remember this section's name for \nameref:

```
4068 \IfValueTF{#3}{%
4069   \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4070   \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4071 }{}}
```

Print an opening comment with the level and the name; ex: "section" "Introduction"

```
4072
4073 \ifbool{HTMLDebugComments}{%
4074   \LWR@htmlcomment{Opening #4 ``#3''{}}%
4075 }{%
4076 }
```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
4077 \ifthenelse{%
4078   \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4079 }%
4080 {\LWR@startpars}%
4081 }%
```

Create the opening tag with an autosec:

```
4082 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

Check if starred:

```
4083 \IfBooleanTF{#1}{%
4084 { %
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
4085 \ifbool{LWR@forcinghtmtoc}{%
4086 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}}%
```

```
4087 { }%
4088 }% starred
```

Not starred, so step counter and add to TOC:

```
4089 { % not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
4090     \ifthenelse{%
4091         \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}}{%
4092     }%
4093     { % if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For article class, lwarp assumes that all is mainmatter.

```
4094     \LWR@traceinfo{LWR@section: about to test main matter}%
4095     \ifbool{LWR@mainmatter}{%
4096     }%
4097     \LWR@traceinfo{LWR@section: yes mainmatter}%
4098     \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
4099     \LWR@traceinfo{LWR@section: about to addcontentsline}%
4100     \addcontentsline{toc}{#4}%
4101     { %
4102         \protect\numberline{\csuse{the#4}}%
4103         \ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4104     }%
4105     \LWR@traceinfo{LWR@section: finished addcontentsline}%
4106     }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
4107     { % not main matter
4108         \LWR@traceinfo{LWR@section: no main matter}%
4109         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4110     }% end of not main matter
4111     }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
4112     { %
4113         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4114     }%
```

For part, print the section type:

```

4115      \ifbool{LWR@mainmatter}%
4116      {%
4117          \ifthenelse{%
4118              \cnttest{\csuse{LWR@depth#4}}{<=}{%
4119                  \value{secnumdepth}}\ AND%
4120              \cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\}%
4121          }%
4122          {\csuse{#4name}~{}{}}%
4123      }%

```

Print the section number:

```

4124      \LWR@traceinfo{LWR@section: about to print section number}%
4125      \ifthenelse{%
4126          \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}\}%
4127      }%
4128      {\protect\LWR@sectionnumber{\seccntformat{#4}}\}%
4129      }%
4130      \LWR@traceinfo{LWR@section: finished print section number}%
4131  }{%
4132 }% end of not starred

```

Print the section name:

```
4133 #3%
```

Close the heading tag, such as /H2:

```
4134 \LWR@htmlltag{\csuse{LWR@tag#4end}}%
```

Generate a L^AT_EX label:

```
4135 \label{autopage-\thepage}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

4136 \ifthenelse{%
4137     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}\}%
4138 }%
4139 {\LWR@startpars}%
4140 }%

```

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.

```
4141 \ifthenelse{%
4142     \NOT\equal{#1}{\BooleanTrue}\OR%
4143     \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
4144 }%
4145 {%
4146     \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4147 }%
4148 {}%
```

Reset to defaults if not a phantomsection:

```
4149 \ifstrempty{#3}%
4150 {}%
4151 {%
4152 \global\boolfalse{LWR@forcinghtmlpage}%
4153 \global\boolfalse{LWR@forcinghtmltoc}%
4154 }%
4155 %
4156 }%
4157 \LWR@traceinfo{LWR@section: done}%
4158 }
```

48.4 \section and friends

```
\part * [<TOC name>] {<name>}
```

```
4159 \DeclareDocumentCommand{\part}{s o m}{%
4160 \LWR@epubprintpendingfootnotes%
4161 \LWR@stopars%
4162 %
4163 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4164 %
4165 \LWR@section{#1}{#2}{#3}{part}%
4166 }
```

```
\chapter * [<TOC name>] {<name>}
```

```
4167 \@ifundefined{chapter}%
4168 {}
```

```
4169 {%
4170 \DeclareDocumentCommand{\chapter}{s o m}{%
4171 \LWR@traceinfo{chapter #3}%
4172 \LWR@epubprintpendingfootnotes%
4173 \LWR@stoppars%
4174
4175 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4176
4177 \LWR@section{#1}{#2}{#3}{chapter}%
4178 }
4179 }

\section * [<TOC name>] {<name>}

4180 \DeclareDocumentCommand{\section}{s o m}{%
4181 \LWR@epubprintpendingfootnotes%
4182 \LWR@stoppars%
4183
4184 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4185
4186 \LWR@section{#1}{#2}{#3}{section}%
4187 }

\subsection * [<TOC name>] {<name>}

4188 \DeclareDocumentCommand{\subsection}{s o m}{%
4189 \LWR@epubprintpendingfootnotes%
4190 \LWR@stoppars%
4191
4192 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4193
4194 \LWR@section{#1}{#2}{#3}{subsection}%
4195 }

\subsubsection * [<TOC name>] {<name>}

4196 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4197 \LWR@epubprintpendingfootnotes%
4198 \LWR@stoppars%
4199
4200 \LWR@startnewdepth{\LWR@depthsubsubsection}%
4201 {\LWR@printclosesubsubsection}%
4202
4203 \LWR@section{#1}{#2}{#3}{subsubsection}%
4204 }
```

```
\paragraph * [<TOC name>] {<name>}
4205 \DeclareDocumentCommand{\paragraph}{s o m}{%
4206 \LWR@epubprintpendingfootnotes%
4207 \LWR@stoppars%
4208
4209 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4210
4211 \LWR@section{#1}{#2}{#3}{paragraph}%
4212 }

\subparagraph * [<TOC name>] {<name>}
4213 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4214 \LWR@epubprintpendingfootnotes%
4215 \LWR@stoppars%
4216
4217 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4218
4219 \LWR@section{#1}{#2}{#3}{subparagraph}%
4220 }

4221 \end{warpHTML}
```

49 Starting a new file

for HTML & PRINT: 4222 \begin{warpall}

\HTMLLanguage Default language for the HTML lang tag.

```
4223 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4224
4225 \newcommand*{\HTMLLanguage}[1]{%
4226 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4227 }

4228 \end{warpall}
```

for HTML output: 4229 \begin{warpHTML}

\LWR@filestart {<title_suffix>}

Creates the opening HTML tags.

```
4230 \newcommand*{\LWR@filestart}[1]{  
4231 \LWR@traceinfo{\LWR@filestart}}
```

Locally temporarily disable direct-formatting commands:

```
4232 \begingroup  
4233 \LWR@traceinfo{\LWR@filestart: A}  
4234 \LWR@nullfonts  
4235 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
4236 \LWR@htmlltag{!DOCTYPE html}\LWR@orignewline  
4237 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
4238 \LWR@htmlltag{html lang="\LWR@currentHTMLLanguage"{} }\LWR@orignewline
```

Start of the meta data:

```
4239 \LWR@htmlltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4240 \LWR@htmlltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
4241 \ifcsempty{\theHTMLAuthor}{}{  
4242 \LWR@htmlltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline  
4243 }
```

lwarf is the generator:

```
4244 \LWR@htmlltag{meta name="generator" content="LaTeX lwarf package" /}%  
4245 \LWR@orignewline
```

If there is a description, add it now:

```
4246 \ifdefempty{\LWR@currentHTMLDescription}{}{  
4247 \LWR@htmlltag{  
4248 meta name="description" content="\LWR@currentHTMLDescription" /}%  
4249 \LWR@orignewline  
4250 }%
```

Mobile-friendly viewport:

```
4251 \LWR@htmltag{meta name="viewport" %
4252 content="width=device-width, initial-scale=1.0" /}%
4253     \LWR@orignewline
```

IE patch:

```
4254 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
4255 \LWR@htmltag{%
4256 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
4257 \LWR@htmltag{/script}\LWR@orignewline
4258 \LWR@htmltag{! [endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4259 \ifcvoid{thetitle}{}{%
4260 \LWR@htmltag{title}\thetitle#\LWR@htmltag{/title}\LWR@orignewline%
4261 }%
```

The page's stylesheet:

```
4262 \LWR@htmltag{%
4263 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4264 \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.

```
4265 \ifbool{mathjax}%
4266 {%
4267     \boolfalse{LWR@verbtags}
4268     \verbatiminput{lwarp_mathjax.txt}%
4269     \booltrue{LWR@verbtags}
4270     \LWR@stoppars
4271 }% end of mathjax
4272 {}%
```

End of the header:

```
4273 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4274 \LWR@htmltag{body}\LWR@orignewline
4275 \endgroup
4276 \LWR@traceinfo{LWR@filestart: done}
4277 }

4278 \end{warpHTML}
```

50 Starting HTML output

for HTML output: 4279 \begin{warpHTML}

\LWR@LwarpStart Executed at the beginning of the entire document.

```
4280 \catcode`$=\active
4281 \newcommand*{\LWR@LwarpStart}{%
4282 {%
4283 \LWR@traceinfo{\LWR@l warpStart}}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
4284 \ifbool{FormatWordProcessor}{%
4285     \setcounter{FileDepth}{-5}%
4286     \boolfalse{HTMLDebugComments}}%
4287 }{}}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
4288 \edef\LWR@strresult{\HomeHTMLFilename}
4289 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4290 \edef\LWR@strresult{\HTMLFilename}
4291 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
4292 \LWR@origonecolumn%
4293 \LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
4294 \LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
4295 \LWR@origraggedright%
4296 \LetLtxMacro{\ }{\LWR@endofline}%
```

Spread the lines for pdftotext to read them well:

```
4297 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4298 \setlength{\parindent}{0pt}
4299 \setlength{\parskip}{2ex}
```

For the `lateximages` record file:

```
4300 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
4301 \setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
4302 \renewcommand{\ps@plain}{}{}
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 4303 \renewcommand*{\centering}{}{}
4304 \renewcommand*{\raggedleft}{}{}
4305 \renewcommand*{\raggedright}{}{}
```

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:

```
4306 \LetLtxMacro{\LWR@origtabular}{\tabular}
4307 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
4308 \LetLtxMacro{\tabular}{\LWR@tabular}
4309 \LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
4310 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

[Label in HTML](#)

```
4311 \let\LWR@origltx@label\ltx@label
4312 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
4313 \let\TeX\LWR@TeX
4314 \let\LaTeX\LWR@LaTeX
4315 \let\LuaTeX\LWR@LuaTeX
4316 \let\LuaLaTeX\LWR@LuaLaTeX
4317 \let\XeTeX\LWR@XeTeX
```

```
4318 \let\XeLaTeX\LWR@XeLaTeX
4319 \let\ConTeXt\LWR@ConTeXt
```

Graphics:

```
4320 \LetLtxMacro{\rotatebox}{\LWR@rotatebox}
4321 \LetLtxMacro{\scalebox}{\LWR@scalebox}
4322 \let\reflectbox\LWR@reflectbox
```

Not yet started any paragraph handling:

```
4323 \global\boolfalse{LWR@doingapar}
4324 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
4325 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4326 \LWR@filestart{}
4327 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4328 \LWR@htmltag{header}\LWR@orignewline
4329 \LWR@startpars
4330 \LWR@firstpagetop
4331 \LWR@stoppars
4332 \LWR@htmltag{/header}\LWR@orignewline
4333 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4334 \LWR@htmltag{section class="textbody"{}}
```

Document and page settings:

```
4335 \mainmatter
4336 \LWR@origpagenumbering{arabic}
```

Patch the itemize, enumerate, and description environments and \item. This works with the native L^AT_EX environments, as well as those provided by enumitem, enumerate, and paralist.

```
4337 \LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
4338 \catcode`$=\active
```

Allow HTML paragraphs to begin:

```
4339 \LWR@startpars
4340 \LWR@traceinfo{LWR@lwarpStart: done}
4341 }
4342 \catcode`$=3% math shift until lwarp starts
```

```
4343 \end{warpHTML}
```

51 Ending HTML output

for HTML output: 4344 \begin{warpHTML}

\LWR@requesttoc {<boolean>} {<suffix>} Requests that a toc, lof, or lot be generated.

```
4345 \newcommand*{\LWR@requesttoc}[2]{%
4346 \ifbool{#1}{%
4347 {
4348   \expandafter\newwrite\csuse{tf@#2}%
4349   \immediate\openout \csuse{tf@#2} \jobname.#2\relax
4350 }{}%
4351 }
```

\LWR@LwarpEnd Final stop of all HTML output:

```
4352 \newcommand*{\LWR@LwarpEnd}{%
4353 {
4354 \LWR@stopars
4355 \LWR@closeprevious{\LWR@depthfinished}}
```

At the bottom of the ending file:

Close the textbody:

```
4356 \LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
4357 \LWR@printpendingfootnotes
```

Create the footer:

```
4358 \LWR@htmlelement{footer}
4359
4360 \LWR@pagebottom
4361
4362 \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.

```
4363 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4364 {}
4365 {
4366     \ifnumcomp{\value{LWR@htmfilename}}{>}{0}{\LWR@botnavigation}{}
4367 }

4368 \LWR@stopars% final stop of all paragraphs
```

Finish the HTML file:

```
4369 \LWR@htmllag{/body}\LWR@orignewline
4370 \LWR@htmllag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4371 \LWR@orignewpage
```

For `\lateximage` commands:

```
4372 \immediate\closeout\LWR@lateximagesfile
4373 }

4374 \end{warpHTML}
```

52 Title page

package support lwarp supports the native L^AT_EX 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 52.7.

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 L^AT_EX 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.

52.1 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 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** `\@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** `\printtitle, etc.` are user-level macros intended to be used in custom `titlepage` or `titlingpage` environments in cases where `\maketitle` is not desired. These `\printdate` commands preserve the `\thanks, etc.,` and should not be used in the main text.
- \thetitle** `\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 L^AT_EX 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.

52.2 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: 4375 `\begin{warpall}`
4376 `\newrobustcmd{\affiliation}[1]{}`
4377 `\end{warpall}`

for PRINT output: 4378 `\begin{warpprint}`

4379 `\AtBeginEnvironment{titlepage}{`
4380 `\renewrobustcmd{\affiliation}[1]{\textsc{\small#1}}`
4381 `}`
4382
4383 `\AtBeginDocument{`
4384 `@ifpackageloaded{titling}{`
4385 `\AtBeginEnvironment{titlingpage}{`
4386 `\renewrobustcmd{\affiliation}[1]{\textsc{\small#1}}`
4387 `}`
4388 `}{}% titling loaded`
4389 `}% AtBeginDocument`

4390 `\end{warpprint}`

for HTML output: 4391 `\begin{warpHTML}`

Env `titlepage` Sets up a `<div>` of class `titlepage`.

```

4392 \renewenvironment*{titlepage}
4393 {
4394 \renewrobustcmd{\affiliation}[1]{\InlineClass{affiliation}{##1}}
4395 \LWR@printpendingfootnotes
4396 \LWR@forcenewpage
4397 \BlockClass{titlepage}
4398 }
4399 {
4400 \endBlockClass
4401 \LWR@printpendingfootnotes
4402 }

4403 \end{warpHTML}

```

52.3 Printing the thanks

for HTML & PRINT: 4404 `\begin{warpall}`

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

```

4405 \newcommand*{\printthanks}{\@thanks}
4406 \end{warpall}

```

52.4 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: 4407 `\begin{warpHTML}`

```

\printtitle
4408 \newcommand*{\printtitle}
4409 {
4410 \LWR@stopars
4411 \LWR@htmlltag{h1}%
4412 \@title%
4413 \LWR@htmlltag{/h1}
4414 \LWR@startpars
4415 }

```

\LWR@printthetitle A private version which prints the title without footnotes, used to title each HTML page.

```
4416 \newcommand*\LWR@printthetitle{  
4417 {  
4418 \LWR@stoppars  
4419 \LWR@htmlltag{h1}%  
4420 \thetitle%  
4421 \LWR@htmlltag{/h1}  
4422 \LWR@startpars  
4423 }
```

\printauthor HTML version.

```
4424 \newcommand*\printauthor{
```

The entire author block is contained in a <div> named author:

```
4425 \begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
4426 \renewcommand{\and}{%  
4427 \end{BlockClass}  
4428 \begin{BlockClass}{oneauthor}  
4429 }
```

Individual authors are contained in a <div> named oneauthor:

```
4430 \begin{BlockClass}{oneauthor}  
4431 @author  
4432 \end{BlockClass}  
4433 \end{BlockClass}  
4434 }
```

\printdate

```
4435 \newcommand*\printdate{  
4436 \begin{BlockClass}{titledate}  
4437 @date  
4438 \end{BlockClass}  
4439 }
```

```
4440 \end{warpHTML}
```

52.5 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: 4441 `\begin{warpprint}`

`\printtitle`

4442 `\newcommand*{\printtitle}{\Huge\titler}`

`\printauthor` Print mode.

4443 `\newcommand*{\printauthor}`

4444 `{\large\begin{tabular}[t]{c}\author\end{tabular}}`

`\printdate`

4445 `\newcommand*{\printdate}{\small\textrit{\@date}}`

4446 `\end{warpprint}`

52.6 \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 45 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4447 `\begin{warpHTML}`

4448 `\if@titlepage`

4449 `\newcommand{\LWR@setfootnoteseries}{%`

4450 `\renewcommand{\thefootnote}{\arabic{c}@\footnote}{%`

4451 `}`

4452 `\else`

4453 `\newcommand{\LWR@setfootnoteseries}{%`

4454 `\renewcommand{\thefootnote}{\fnsymbol{c}@\footnote}{%`

4455 `}`

4456 `\fi`

\LWR@maketitlesetup Patches \thanks macros.

```
4457 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
4458 \LWR@setfootnoteseries%
4459 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

```
\thefootnote \Rightarrow \nameuse{arabic}{footnote}, or
\thefootnote \Rightarrow \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
4460 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4461 \textsuperscript{\@thefnmark} \LWR@orighspace{1in}
\makethanksmark \Rightarrow \thanksfootmark \Rightarrow \tamark \Rightarrow
\@thefnmark \Rightarrow \itshape a (or similar)
```

Print the text:

```
4462 ##1%
4463 }%
4464 }
```

```
\@fnsymbol {\langle counter\rangle}
```

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used \| which was not being seen by pdftotext.

```
4465 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4466 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLunicode{2016}}\or
4467 **\or \HTMLentity{dagger}\HTMLentity{dagger}\or
4468 \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.

```
4469 \renewcommand*{\maketitle}{%
```

An HTML titlepage <div> is used for all classes.

```
4470 \begin{titlepage}
```

Set up special patches:

```
4471 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4472 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4473 \@thanks
```

Close the HTML titlepage div and cleanup:

```
4474 \end{titlepage}
4475 \setcounter{footnote}{0}%
4476 \global\let\thanks\relax
4477 \global\let\maketitle\relax
4478 \global\let \@maketitle\relax
4479 \global\let \@thanks\@empty
4480 \global\let \@author\@empty
4481 \global\let \@date\@empty
4482 \global\let \@title\@empty
4483 \global\let \@title\relax
4484 \global\let \@author\relax
4485 \global\let \@date\relax
4486 \global\let \and\relax
4487 }
```

\@maketitle HTML mode. Typesets the title, etc.:

```
4488 \DeclareDocumentCommand{\@maketitle}{}{%
4489     \LWR@stopars\LWR@htmltag{h1}
4490     \@title
4491     \LWR@htmltag{/h1}\LWR@startpars
4492     \begin{BlockClass}{author}
4493         \renewcommand{\and}{%
4494             \end{BlockClass}
4495             \begin{BlockClass}{oneauthor}
4496         }
4497         \begin{BlockClass}{oneauthor}
4498             \@author
4499             \end{BlockClass}
4500     \end{BlockClass}}
```

```
4501     \begin{BlockClass}{titledate}
4502     \@date
4503     \end{BlockClass}
4504 }
```

\LWR@titlingmaketitle \maketitle for use inside an `HTML titlingpage` environment.

```
4505 \newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
4506 \@thanks
```

Set up special patches:

```
4507 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4508 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4509 \@thanks
4510 }
```

```
4511 \end{warpHTML}
```

52.7 \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.

After `titling` is loaded, `\AddSubtitlePublished` is created, which when used then creates a number of additional macros, and also assigns some of the `titling` hooks.

⚠ 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: 4512 `\begin{warpall}`

\AddSubtitlePublished Adds \published and \subtitle, and related.

```

4513 \AfterPackage{titling}%
4514 \newcommand*\AddSubtitlePublished{%
4515
4516 \newcommand{\@published}{}%
4517
4518 \newcommand{\published}[1]{\gdef\@published{##1}}%
4519
4520 \renewcommand*{\maketitlehooka}{\printpublished}%
4521
4522 \newcommand*{\printpublished}{%
4523 \warpprintonly{\begin{center}\unskip}%
4524 \begin{BlockClass}{published}%
4525 \warpprintonly{\large\itshape}%
4526 \@published%
4527 \end{BlockClass}%
4528 \warpprintonly{\end{center}}%
4529 }%
4530
4531 \newcommand{\@subtitle}{}%
4532
4533 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}%
4534
4535 \renewcommand*{\maketitlehookb}{\prints_subtitle}%
4536
4537 \newcommand*{\prints_subtitle}{%
4538 \warpprintonly{\begin{center}\unskip}%
4539 \begin{BlockClass}{subtitle}%
4540 \warpprintonly{\Large\itshape}%
4541 \@subtitle%
4542 \end{BlockClass}%
4543 \warpprintonly{\end{center}}%
4544 }%
4545
4546 }% \AddSubtitlePublished
4547 }% AfterPackage

4548 \end{warpall}

```

53 Abstract

The following code replaces the L^AT_EX default, and will itself be replaced later if the abstract package is loaded.

for HTML output: 4549 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.
Also over-written by the babel package.

```
4550 \providecommand*\{\abstractname\}{Abstract}

Env abstract

4551 \DeclareDocumentEnvironment{abstract}{}%
4552 {
4553 \LWR@forcenewpage
4554 \BlockClass{abstract}
4555 \BlockClassSingle{abstracttitle}{\abstractname}
4556 }
4557 {
4558 \endBlockClass
4559 }
```

4560 \end{warpHTML}

54 Quote and verse

54.1 Citations and attributions

\attribution for use inside quote, quotation, verse:

ex: \attribution{author name} --- \citetitle{book name}

for HTML output: 4561 \begin{warpHTML}
4562 \newcommand{\attribution}[1]{%
4563 \InlineClass{\attribution}{--\,\#1}}% emdash
4564 \end{warpHTML}

for PRINT output: 4565 \begin{warpprint}
4566 \newcommand{\attribution}[1]{\textsc{---\,\#1}}
4567 \end{warpprint}

\citetitle for use inside quote, quotation, verse:

for HTML output: 4568 \begin{warpHTML}
4569 \newcommand{\citetitle}[1]{%
4570 \InlineClass{\citetitle}{--\,\#1}}% emdash
4571 \end{warpHTML}

```
for PRINT output: 4572 \begin{warpprint}
4573 \newcommand{\citetitle}[1]{\textsf{---},#1}
4574 \end{warpprint}
```

54.2 Quotes, quotations

for HTML output: 4575 \begin{warpHTML}

```
Env quote
4576 \renewenvironment*{quote}
4577 {
4578 \LWR@forcenewpage
4579 \LWR@htmlblocktag{blockquote}
4580 }
4581 {\LWR@htmlblocktag{/blockquote}}
4582
4583 \renewenvironment*{quotation}
4584 {
4585 \LWR@forcenewpage
4586 \LWR@htmlblocktag{blockquotation}
4587 }
4588 {\LWR@htmlblocktag{/blockquotation}}
4589 \end{warpHTML}
```

54.3 Verse

\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 \leftskip` These lengths are used by `verse` and `memoir` to control the left margin, and they
`Len \leftmargini` may already be set by the user for print output. New lengths `\HTMLvleftskip` and
`Len \TMLvleftskip`
`Len \TMLleftmargini`

\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.

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.

for HTML & PRINT: 4590 \begin{warpall}

The following lengths 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 \TMLvleftskip Sets \vleftskip inside a `verse` environment in HTML.

```
4591 \newlength{\HTMLvleftskip}
4592 \setlength{\HTMLvleftskip}{1em}
```

Len \TMLleftmargini Sets \leftmargini inside a `verse` environment in HTML.

```
4593 \newlength{\HTMLleftmargini}
4594 \setlength{\HTMLleftmargini}{4.5em}
```

```
4595 \end{warpall}
```

55 Verbatim

for HTML & PRINT: 4596 \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.

```
4597 \newlength{\VerbatimHTMLWidth}
4598 \setlength{\VerbatimHTMLWidth}{4in}
4599 \end{warpall}
```

for HTML output: 4600 \begin{warpHTML}

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing \verbatiminput in the HTML head.

```
4601 \newbool{LWR@verbtags}
4602 \booltrue{LWR@verbtags}
```

\LWR@atbeginverbatim [⟨style⟩] {⟨class⟩}

Encloses a verbatim environment with the given css class.

```
4603 \newcommand*{\LWR@atbeginverbatim}[2] []
4604 {%
```

Avoid excessive space between lines:

```
4605 \setlength{\parskip}{0ex} %
```

Stop generating HTML paragraph tags:

```
4606 \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.

```
4607 \ifbool{LWR@verbtags}{%
4608     \LWR@htmltag{pre class="#2"
4609     \ifthenelse{\equal{#1}{}}{}{style="#1"}}
4610 }% pre
4611 }{}%
```

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.

Also turn off babel-french extra space before punctuation:

```
4612 \begingroup%
4613 \LWR@origtfamily%
4614 \LWR@FBcancel%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain L^AT_EX \hspace functionality:

```
4615 \LetLtxMacro{\hspace}{\LWR@orighspace}%
4616 }
```

\LWR@afterendverbatim Finishes enclosing a verbatim environment.

```
4617 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the pre:

```
4618 \endgroup%
4619 \unskip%
```

At the end of the environment, close the pre:

```
4620 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
4621
4622 }{}%
```

Resume regular paragraph handling:

```
4623 \LWR@startpars%
4624 }
```

\verbatiminput {<filename>}

Patch \verbatiminput to add HTML tags:

```
4625 \let\LWRV@origverbatim@input\verbatim@input
4626
4627 \renewcommand{\verbatim@input}[2]{%
4628 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
4629 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
4630 \LWRV@origverbatim@input{\#1}{\#2}%
4631 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4632 }
```

Env verbatim

```
4633 \AfterEndPreamble{
4634 \LWR@traceinfo{Patching verbatim.}
4635 \AtBeginEnvironment{verbatim}{%
4636 \LWR@forcenewpage
4637 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4638 }
4639 \AfterEndEnvironment{verbatim}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
4640 }

4641 \end{warpHTML}
```

56 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 of class theoremlabel.
- The contents are placed inside a <div> of class theoremcontents.

```
for HTML output: 4642 \begin{warpHTML}

\@begintheorem {\langle name\rangle} {\langle number\rangle}

4643 \renewcommand{\@begintheorem}[2]{%
4644 \LWR@forcenewpage
4645 \BlockClass{theoremcontents}
4646 \InlineClass{theoremlabel}{#1\ #2\ }%
4647 }

\@opargbegintheorem {\langle name\rangle} {\langle number\rangle} {\langle oparg\rangle}

4648 \renewcommand{\@opargbegintheorem}[3]{%
4649 \LWR@forcenewpage
4650 \BlockClass{theoremcontents}
4651 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }%
4652 }

\@endtheorem

4653 \renewcommand*{\@endtheorem}{%
4654 \endBlockClass% theoremcontents
4655 }

4656 \end{warpHTML}
```

57 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}
```

57.1 Itemize

for HTML output: 4657 `\begin{warpHTML}`

`\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.

```
4658 \newcommand*{\LWR@itemizeitem}{%
4659 \LWR@stoppars%
4660 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem{}}%
4661 \LWR@htmlltag{li}%
4662 \LWR@startpars%
4663 \LWR@origitem%
4664 }
```

Env `itemize [<options>]`

```
4665 \newcommand*{\LWR@itemizestart}{%
4666 \LWR@stoppars%
4667 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4668 \LWR@htmlltag{ul style="list-style-type:none"{} }%
4669 \LWR@startpars%
4670 \let\item\LWR@itemizeitem%
4671 }
4672
4673 \newcommand*{\LWR@itemizeend}{%
```

```

4674 \LWR@stoppars%
4675 \LWR@closeprevious{\LWR@depthlistitem}%
4676 \LWR@closeoneprevious{}%
4677 \LWR@startpars%
4678 }

```

57.2 Enumerate

An HTML unordered list is used with customized L^AT_EX-generated labels.

Env `enumerate` [*<options>*]

```

4679 \newcommand*\LWR@enumeratestart}{%
4680 \LWR@stoppars%
4681 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4682 \LWR@htmlltag{ul style="list-style-type:none"{} }%
4683 \LWR@startpars%
4684 \let\item\LWR@itemizeitem%
4685 }
4686
4687 \newcommand*\LWR@enumerateend}{%
4688 \LWR@stoppars%
4689 \LWR@closeprevious{\LWR@depthlistitem}%
4690 \LWR@closeoneprevious{}%
4691 \LWR@startpars%
4692 }

```

57.3 Description

`\LWR@descitem` [*<label>*] Handles an `\item` inside a description.

```

4693 \newcommand*\LWR@descitem}[1] []%
4694 {%
4695 \LWR@stoppars%
4696 \LWR@setlatestname{#1}%
4697 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%

```

Temporarily disable `\hspace`, which `article.cls`, etc. use per `\item` for descriptions only. This causes `l warp` to mistakenly place an empty span between HTML list tags.

```
4698 \LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original \item code:

```
4699 \LWR@origitem[]%
```

Restore \hspace for use in the item text:

```
4700 \LetLtxMacro{\hspace}{\LWR@hspace}%
```

Be sure the label doesn't print to the left of the rest of the file:

```
4701 \LWR@orighspace{in}
4702 \LWR@htmlltag{dt}#1\LWR@htmlltag{/dt}%
4703 \LWR@orignewline%
4704 \LWR@htmlltag{dd}%
4705 \LWR@startpars%
4706 }
```

Env description [⟨options⟩]

```
4707 \newcommand*{\LWR@descriptionstart}{%
4708 \LWR@stoppars%
4709 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4710 \LWR@htmlltag{dl}%
4711 \LWR@startpars%
4712 \let\item\LWR@descitem%
4713 }
4714
4715 \newcommand*{\LWR@descriptionend}{%
4716 \LWR@stoppars%
4717 \LWR@closeprevious{\LWR@depthlistitem}%
4718 \LWR@closeoneprevious{}%
4719 \LWR@startpars%
4720 }
```

\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 L^AT_EX environments, as well as those provided by enumitem, enumerate, and paralist.

```
4721 \newcommand*{\LWR@patchlists}{%
4722 \LetLtxMacro{\LWR@origitem}{\item}
4723 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4724 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4725 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4726 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
```

```

4727 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4728 \AtEndEnvironment{description}{\LWR@descriptionend}
4729 }

4730 \end{warpHTML}

```

58 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

58.1 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 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 ignored in print mode.

```

\StartDefiningTabulars
<define macros or environments using tabular and &
here>
\EndDefiningTabulars

```

⚠ 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 environment 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}
{
\begin{tabular}{cc}
left & right \\
\end{tabular}
}
{
\TabularMacro\ResumeTabular
left & right \\
\end{tabular}
}
\EndDefiningTabulars
```

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.
- 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.

⚠️ \multicolumn & \multirow

- lwarp does not support combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow[2]{c}[c]{3}[0]{in}[opt]{Text}
```

The two arguments for \multicolumn come first, followed by the five arguments for \multirow, many of which are optional, followed by the contents. See section 168.2.

- 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 types**

- Vertical rules are not yet supported.

- * in a column specification is not used (so far). Repeat the column type the correct number of times.

- Only one each of @, !, >, and < may be used at each column, and they are used in that order.

- \newcolumntype is ignored; unknown column types are set to 1.

⚠️ * column specification

Rules:

- If a multirow reaches to the bottom of a table, and \bottomrule does not go over to that edge, try adding a line of empty cells below the \bottomrule. (This may be a browser bug.)
 - If a \midrule is desired after the last row, an additional row of blank cells must be used.
 - \cmidrule does not support width or trim options due to css limitations.
- \multirule with rules
rule at last row
- \cmidrule width, trim
 \cmidrule and
 \multicolumn
- ⚠ \warpprintonly
misplaced \noalign

• \cmidrule borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has \multicolumn cells below \cmidrule borders.

- 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 relevant to HTML output.

Other:

longtable headings

⚠ S columns

- tabularx ignores the width, but X columns do produce paragraph columns or multicolumns.

- For longtable, place headings and footings which do not apply to HTML inside \warpprintonly{}.

- 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 \\}
```

58.2 Token lookahead

Used by \LWR@futureonospacelet to look at the next token.

for HTML output: 4731 \begin{warpHTML}

```
\LWR@mynexttoken
```

```
4732 \newcommand{\LWR@mynexttoken}{\relax}
```

\futurelet copies the next token then executes a function to analyze

\LWR@futurenonospacelet does the same, but ignores intervening white space

Based on the booktabs style:

```
\LWR@futurenonospacelet
```

```
4733 \def{\LWR@futurenonospacelet#1{\def{\LWR@cs{#1}}%  
4734 \afterassignment{\LWR@fnalone\let\nexttoken= }  
4735 \def{\LWR@fnalone{\expandafter{\futurelet{\LWR@cs{\LWR@fnltwo}}%  
4736 \def{\LWR@fnltwo{\%  
4737 \expandafter{\ifx{\LWR@cs{\@spoken}\let\next=\LWR@fnlthree}}%  
4738 \else\let\next=\nexttoken\fi\next}  
4739 \def{\LWR@fnlthree{\afterassignment{\LWR@fnalone\let\next= }}
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
4740 \newcommand*{\LWR@getmynexttoken}{%  
4741 \LWR@traceinfo{\LWR@getmynexttoken}%  
4742 % nothing must follow this next line  
4743 \LWR@futurenonospacelet{\LWR@mynexttoken{\LWR@tabledatacolumntag}}  
4744 }
```

58.3 Booleans

Bool LWR@startedrow True if should print a row tag before this column.

```
4745 \newbool{\LWR@startedrow}  
4746 \boolfalse{\LWR@startedrow}
```

Bool LWR@doinghline True if the next row will have an hline above it.

```
4747 \newbool{\LWR@doinghline}  
4748 \boolfalse{\LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
4749 \newbool{\LWR@doingtbrule}  
4750 \boolfalse{\LWR@doingtbrule}
```

Bool LWR@tableparcell True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
4751 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
4752 \newbool{LWR@skippingmrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not print the trailing @ or ! columns.

```
4753 \newbool{LWR@skipatbang}
```

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.

```
4754 \newbool{LWR@intabularmetadata}
```

```
4755 \boolfalse{LWR@intabularmetadata}
```

58.4 Handling &, @, and !

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/>
[where-do-i-find-futurelets-nasty-behaviour Documented/11860#11860](http://tex.stackexchange.com/questions/11860/#11860)

\LWR@closetabledatacell If LWR@skippingmrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
4756 \newcommand*\LWR@closetabledatacell{%
4757 \global\booltrue{LWR@intabularmetadata}%
4758 \ifbool{LWR@existingtabular}{}{%
4759 {}% not exiting tabular
4760   \ifbool{LWR@skippingmrowcell}{}{%
4761     {}% not skippingmrowcell
4762     \unskip%
4763     \LWR@getexpparray{LWR@col afterspec}{\theLWR@tablecolspos}%
4764 % % \LWR@getexpparray{LWR@col at spec}{\theLWR@tablecolspos}%
4765 % \LWR@printatbang{at}{\theLWR@tablecolspos}%
4766 % % \LWR@getexpparray{LWR@col bang spec}{\theLWR@tablecolspos}%
4767 % \LWR@printatbang{bang}{\theLWR@tablecolspos}%
4768 }
```

Insert any < then any @ and ! column contents:

```
4762 \unskip%
4763 \LWR@getexpparray{LWR@col afterspec}{\theLWR@tablecolspos}%
4764 % % \LWR@getexpparray{LWR@col at spec}{\theLWR@tablecolspos}%
4765 % \LWR@printatbang{at}{\theLWR@tablecolspos}%
4766 % % \LWR@getexpparray{LWR@col bang spec}{\theLWR@tablecolspos}%
4767 % \LWR@printatbang{bang}{\theLWR@tablecolspos}%
4768 }
```

Close paragraphs:

```
4768 \ifbooleq{\LWR@tableparcell}{\LWR@stoppars}{}%
4769 \global\boolfalse{\LWR@tableparcell}%
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```
4770 \leavevmode\unskip\LWR@htmtag{/td}\LWR@orignewline%
4771 \ifbooleq{\LWR@skipatbang}%
4772 {\boolfalse{\LWR@skipatbang}}%
4773 {%
4774     \LWR@printatbang{at}{\theLWR@tablecolspos}%
4775     \LWR@printatbang{bang}{\theLWR@tablecolspos}%
4776 }% not skipping at or bang
4777 }% not skipping mrowcell
4778 }% not exiting tabular
4779 \global\boolfalse{\LWR@skippingmrowcell}%
4780 }
```

`LWR@tabulardepth` tracks whether & is being used inside a `tabular`.

```
4781 \newcounter{\LWR@tabulardepth}
4782 \setcounter{\LWR@tabulardepth}{0}
4783
```

When not used inside a `tabular`, & performs its original function as recorded here (with catcode 4).

```
4784 \let\LWR@origampmacro&
4785 \end{warpHTML}
```

58.4.1 Localizing & catcodes

for HTML & PRINT: 4786 `\begin{warpall}`

⚠ **misplaced alignment tab character &** Place `\StartDefiningTabulars` and `\EndDefiningTabulars` 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.

```
4787 \newcommand{\StartDefiningTabulars}{%
4788 \LWR@traceinfo{\StartDefiningTabulars}}
```

```
4789 \warpHTMLonly{\catcode`&=\active}%
4790 }
```

\EndDefiningTabulars Place after defining something with & in it.

```
4791 \newcommand{\EndDefiningTabulars}{%
4792 \LWR@traceinfo{EndDefiningTabulars}%
4793 \warpHTMLonly{\catcode`&=4}%
4794 }

4795 \end{warpall}
```

58.4.2 Handling &

for HTML output: 4796 \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.

```
4797 \newcommand*\LWR@tabularampersand{%
4798 \LWR@traceinfo{\LWR@tabularampersand}%
4799 \ifthenelse{\cnttest{\value{\LWR@tabulardepth}}{>}{0}}{%
4800 {%
```

If not skipping a multirow cell, close the current data cell.

```
4801 \unskip%
4802 \LWR@closetabledatacell%
```

Move to the next column.

```
4803 \addtocounter{\LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
4804 \LWR@getmynexttoken%
4805 }%
```

If not inside a tabular, performs the original action:

```
4806 {\LWR@origampmacro}%
4807 }
```

& 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.

58.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

```
\LWR@tabularendofline
```

```
4808 \NewDocumentCommand{\LWR@tabularendofline}{s o}
4809 {%
4810 \LWR@closetabledatacell%
```

Finish the previous row:

```
4811 \LWR@htmlltag{/tr}\LWR@orignewline
4812 \global\booltrue{\LWR@intabularmetadata}
```

Not yet started a table row:

```
4813 \global\boolfalse{\LWR@startedrow}
```

Additional setup:

```
4814 \global\boolfalse{\LWR@doinghline}%
4815 \global\boolfalse{\LWR@doingtbrule}%
4816 \LWR@clearmidrules%
```

Start at first column:

```
4817 \setcounter{\LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```
4818 \LWR@getmynexttoken%
4819 }
```

58.6 Variables

```
4820 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4821 \newcommand*{\LWR@pposition}{}%
4822 \newcommand*{\LWR@pleft}{}%
4823 \newcommand*{\LWR@pright}{}%
```

\LWR@tablecolspec Holds the parsed column specification, of total width LWR@tabletotalcols, not counting @ and ! columns.

Will contain a string such as llrrccpc, exactly one letter per *LATEX* table column, without @, !, >, <, or the vertical pipe.

This is indexed by the counter LWR@tabletotalcols.

```
4824 \newcommand*{\LWR@tablecolspec}{}%
```

\LWR@strresult Holds the result of Str functions.

```
4825 \newcommand*{\LWR@strresult}{}%
4826 \newcommand*{\LWR@strresulttwo}{}%
```

\LWR@origcolspec Holds the original column specs given to tabular.

```
4827 \newcommand*{\LWR@origcolspec}{}%
```

Ctr LWR@tablecolswidth Holds the width of the table columns specification.

This is the number of tokens, including one for each @ etc. column, and also one each for the parameters of p, @, etc. columns, and three for each D column.

(This is not the total # of *LATEX* columns in the table.)

```
4828 \newcounter{LWR@tablecolswidth}
```

Ctr LWR@tablecolspos Where are currently looking into the table column specification. Index starts at 1.

```
4829 \newcounter{LWR@tablecolspos}
```

Ctr LWR@tabletotalcols Holds the final number of table columns, not counting @ and ! columns.

```
4830 \newcounter{LWR@tabletotalcols}
```

Ctr LWR@tabletotalcolsnext Holds the next column while parsing. Is one more than LWR@tabletotalcols.

```
4831 \newcounter{LWR@tabletotalcolsnext}
```

- LWR@colatspec A data array of specifications for @ columns. The leftmost's index is `leftedge`, the others are counter values. See section 32.
- LWR@colbangspec A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 32.
- LWR@colbeforespec A data array of specifications for > columns.
- LWR@colafterspec A data array of specifications for < columns.

58.7 Parsing @, >, <, ! columns

Holds the parsed argument for @, >, <, or ! columns:

```
4832 \newcommand*{\LWR@colparameter}{}%
```

\LWR@parseatcolumn Handles @{text} columns.

```
4833 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
4834 \LWR@traceinfo{@ column}%
4835 \addtocounter{\LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
4836 \LWR@traceinfo{about to read the next token}%
4837 \expandarg%
4838 \StrChar{\LWR@origcols}{\the\LWR@tablecolspos}{\LWR@colparameter}%
4839 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4840 \LWR@traceinfo{have now read the next token}%
4841 \ifthenelse{\cnttest{\value{\LWR@tabletotalcols}}=0}%
4842 {%
4843   \LWR@traceinfo{at the left edge}%
4844   \LWR@setexpparray{\LWR@colatspec}{leftedge}{\LWR@colparameter}%
4845   \LWR@traceinfo{at the left edge: \%}
4846   \LWR@getexpparray{\LWR@colatspec}{leftedge}%
4847 }%
4848 {%
4849   \LWR@traceinfo{not at the left edge}%
4850   \LWR@setexpparray{\LWR@colatspec}{\the\LWR@tabletotalcols}{\LWR@colparameter}%
4851   \LWR@traceinfo{at \the\LWR@tabletotalcols: \%}
```

```

4852     \LWR@getexparray{\LWR@colatspec}{\theLWR@tabletotalcols}}%
4853 }%
4854 \let\LWR@colparameter\relax%
4855 \booltrue{\LWR@validtablecol}%
4856 }

```

\LWR@parsebangcolumn Handles !{text} columns.

```
4857 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```

4858 \LWR@traceinfo{bang column}%
4859 \addtocounter{\LWR@tablecolspos}{1}%

```

Read the next token into \LWR@strresult, expanding once:

```

4860 \LWR@traceinfo{about to read the next token:}%
4861 \expandarg%
4862 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4863 \fullexpandarg%

```

Store the result into a data array, expanding once out of \LWR@strresult:

```

4864 \LWR@traceinfo{have now read the next token}%
4865 \ifthenelse{\cnttest{\value{\LWR@tabletotalcols}}=0}%
4866 {%
4867   \LWR@traceinfo{at the left edge}%
4868   \LWR@setexparray{\LWR@colbangspec}{leftedge}{\LWR@colparameter}%
4869 }%
4870 {%
4871   \LWR@traceinfo{not at the left edge}%
4872   \LWR@setexparray{\LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4873   \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colparameter!}%
4874 }%
4875 \let\LWR@colparameter\relax%
4876 \booltrue{\LWR@validtablecol}%
4877 }

```

\LWR@parsebeforecolumn Handles >{text} columns.

```
4878 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
4879 \addtocounter{\LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```
4880 \expandarg%
4881 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4882 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4883 \LWR@setexpparray[LWR@colbeforespec]{\theLWR@tabletotalcolsnext}{\LWR@colparameter}%
4884 \let\LWR@colparameter\relax%
4885 \booltrue{LWR@validtablecol}%
4886 }
```

`\LWR@parseaftercolumn` Handles <{text} columns.

```
4887 \newcommand*\LWR@parseaftercolumn{%
```

Move to the next token after the '<':

```
4888 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```
4889 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@colparameter]
4890 \expandarg%
4891 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4892 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4893 \LWR@setexpparray[LWR@colaferspec]{\theLWR@tabletotalcols}{\LWR@colparameter}%
4894 \let\LWR@colparameter\relax%
4895 \booltrue{LWR@validtablecol}%
4896 }
```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```
4897 \newcommand*\LWR@parseskipcolumn{%
4898 \booltrue{LWR@validtablecol}%
4899 }%
```

58.8 Parsing 'l', 'c', or 'r' columns

`\LWR@parsonormalcolumn` {<*thiscolumn*>}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
4900 \newcommand*{\LWR@parsenormalcolumn}[1]{%
4901 \appto{\LWR@tablecolspec}{#1}%
4902 \addtocounter{\LWR@tabletotalcols}{1}%
4903 \addtocounter{\LWR@tabletotalcolsnext}{1}%
4904 \LWR@traceinfo{normal column \the\LWR@tabletotalcols: #1}%
4905 \LWR@setexpparray{\LWR@colatspec}{\the\LWR@tabletotalcolsnext}{}%
4906 \LWR@setexpparray{\LWR@colbangspec}{\the\LWR@tabletotalcolsnext}{}%
4907 \LWR@setexpparray{\LWR@colbeforespec}{\the\LWR@tabletotalcolsnext}{}%
4908 \LWR@setexpparray{\LWR@col afterspec}{\the\LWR@tabletotalcolsnext}{}%
4909 \booltrue{\LWR@validtablecol}%
4910 }
```

58.9 Parsing ‘p’, ‘m’, or ‘b’ columns

\LWR@parsepcolumn {*thiscolumn*} The width will be ignored.

```
4911 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
4912 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
4913 \addtocounter{\LWR@tablecolspos}{1}%
4914 }
```

58.10 Parsing ‘D’ columns

From the dcolumn package.

\LWR@parseDcolumn {*thiscolumn*} The three parameters will be ignored.

```
4915 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
4916 \LWR@parsenormalcolumn{#1}%
```

Table 6: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Skips the following three parameters.

```
4917 \addtocounter{LWR@tablecolspos}{3}%
4918 }
```

58.11 Parsing the column specifications



HTML CSS cannot exactly match the L^AT_EX concept of a baseline for a table row. Table 6 shows the L^AT_EX 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 7 for details.

Table 7 describes how each kind of column is converted to HTML.

Bool LWR@validtablecol True if found a valid table column type.

```
4919 \newbool{LWR@validtablecol}
```

\LWR@parsetablecols {<colspecs>}

Scans the column specification left to right.

Builds LWR@tablecolspec with the final specification, one column per entry. The number of final columns is stored in LWR@tabletotalcols.

```
4920 \newcommand*{\LWR@parsetablecols}[1]{%
4921 \LWR@traceinfo{LWR@parsetablecols started}}%
```

Remember the original supplied column spec:

```
4922 \renewcommand*{\LWR@origcolspec}{#1}%
```

Table 7: 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 6, L^AT_EX 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 L^AT_EX 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'. From the `siunitx` package.
- D:** Converted to 'c'. From the `dcolumn` package.
- @, !, >, <:** One each, in that order.
- Unknown:** Converted to 'l'.
- \newcolumn:** Currently treated as unknown.
-

Clear the parsed resulting column spec:

```
4923 \renewcommand*\{LWR@tablecolspec}\{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
4924 \setcounter{LWR@tabletotalcols}{0}%
4925 \setcounter{LWR@tabletotalcolsnext}{1}%
4926 \LWR@setexpparray{LWR@colatspec}{leftedge}{}%
4927 \LWR@setexpparray{LWR@colatspec}{1}{}%
4928 \LWR@setexpparray{LWR@colatspec}{2}{}%
4929 \LWR@setexpparray{LWR@colatspec}{3}{}%
4930 \LWR@setexpparray{LWR@colbangspec}{leftedge}{}%
4931 \LWR@setexpparray{LWR@colbangspec}{1}{}%
4932 \LWR@setexpparray{LWR@colbangspec}{2}{}%
4933 \LWR@setexpparray{LWR@colbangspec}{3}{}%
4934 \LWR@setexpparray{LWR@colbeforespec}{1}{}%
4935 \LWR@setexpparray{LWR@colbeforespec}{2}{}%
4936 \LWR@setexpparray{LWR@colbeforespec}{3}{}%
4937 \LWR@setexpparray{LWR@colafterspec}{1}{}%
4938 \LWR@setexpparray{LWR@colafterspec}{2}{}%
4939 \LWR@setexpparray{LWR@colafterspec}{3}{}%
```

Starting at the first column specification:

```
4940 \setcounter{LWR@tablecolspos}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```
4941 \LWR@traceinfo{about to StrLen}%
4942 \noexpandarg%
4943 \StrLen{\#1}[\LWR@strresult]%
4944 \fullexpandarg%
4945 \LWR@traceinfo{finished StrLen}%
4946 \setcounter{LWR@tablecolswidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
4947 \whiledo{\not\value{LWR@tablecolspos}>\value{LWR@tablecolswidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```
4948 \noexpandarg%
4949 \StrChar{\#1}{\theLWR@tablecolspos}[\LWR@strresult]%
4950 \LWR@traceinfo{position \arabic{LWR@tablecolspos}: \LWR@strresult}%
4951 \fullexpandarg%
```

Not yet found a valid column type

```
4952 \boolfalse{\LWR@validtablecol}%
```

 Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```
4953 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}%
4954 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}%
4955 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}%
4956 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{1}}%
4957 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}%
4958 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}%
4959 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{1}}%
4960 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}%
4961 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}%
4962 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}%
4963 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}%
4964 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}%
4965 \IfStrEq{\LWR@strresult}{|}{\LWR@parsingskipcolumn}%
4966 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}%
4967 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}%
4968 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}%
```

From the dcolumn package:

```
4969 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
4970 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}%
```

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumntype{P}[1]{>{\centering\arraybackslash}p{#1}}
```

```
4971 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}%
4972 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}%
4973 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}%
```

If this column was an invalid column type, convert it to an l column:

```
4974 \ifbool{\LWR@validtablecol}{}{%
4975   \LWR@traceinfo{invalid column type: \LWR@strresult}%
4976   \LWR@parsenormalcolumn{l}%
4977 }%
```

```

4978 \addtocounter{LWR@tablecolspos}{1}%
4979 }%
4980 }%

```

58.12 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.

```

4981 \newcommand*{\LWR@maybenewtablerow}%
4982 {%
4983 \ifbool{LWR@startedrow}%
4984 {}% started the row
4985 {}% not started the row

```

Remember that now have started the row:

```
4986 \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```

4987 \global\booltrue{LWR@intabularmetadata}%
4988 \ifbool{LWR@doinghline}%
4989 {\LWR@htmltag{tr class="hline"{} }\LWR@orignewline}%
4990 {}% not doing hline
4991 \ifbool{LWR@doingtbrule}%
4992 {\LWR@htmltag{tr class="tbrule"{} }\LWR@orignewline}%
4993 {\LWR@htmltag{tr}\LWR@orignewline}%
4994 }% end of not doing hline
4995 }% end of not started the row
4996 }

```

58.13 Printing at or bang tags

```

\LWR@printatbang {\langle at -or- bang\rangle} {\langle index\rangle}

4997 \newcommand*{\LWR@printatbang}[2]{%
4998 \edef\LWR@atbangspec{\LWR@getexparray{LWR@col#1spec}{#2}}%
4999 \LWR@traceinfo{@bang: !\LWR@atbangspec!}%
5000 \ifdefempty{\LWR@atbangspec}%
5001 % \ifthenelse{\isempty{\LWR@atbangspec}}{%
5002 {}%
5003 {}%

```

```

5004     \LWR@htmltag{td class="td#1%
5005     \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}}{\theLWR@tablecolspos}}{Y}{rule}{}%
5006     "}%
5007     \LWR@atbangspec%
5008     \LWR@htmltag{/td}\LWR@orignewline%
5009 }%
5010 }%

```

58.14 Data opening tag

\LWR@tabledatasinglecolumntag Print a table data opening tag with style for alignment

```

5011 \newcommand*{\LWR@tabledatasinglecolumntag}{%
5012 {%
5013 \LWR@maybenewtablerow%

```

Don't start a new paragraph tag if have already started one:

```

5014 \ifbool{\LWR@intabularmetadata}{%
5015 {%

```

If have found the end of tabular command, do not create the next data cell:

```

5016 \ifbool{\LWR@exittingtabular}{%
5017 {%

```

Print the @ and ! contents before first column:

```

5018 \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}{%
5019 {%
5020     \LWR@printatbang{at}{leftedge}%
5021     \LWR@printatbang{bang}{leftedge}%
5022 }% left edge
5023 {}% not left edge

```

Fetch the current column's alignment character into \LWR@strresult:

```

5024 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

5025 \LWR@htmltag{td class="td%

```

append this column's spec:

```

5026 \LWR@strresult%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
5027      \ifthenelse{\equal{\LWR@getexpparray{LWR@midrules}{\theLWR@tablecolspos}}{Y}}%
5028          {rule}%
5029          {}%
5030          "{}"%
```

If this is a p, m, b, or X column, allow paragraphs:

```
5031      \ifthenelse{%
5032          \equal{\LWR@strresult}{p}\OR%
5033          \equal{\LWR@strresult}{m}\OR%
5034          \equal{\LWR@strresult}{b}\OR%
5035          \equal{\LWR@strresult}{P}\OR%
5036          \equal{\LWR@strresult}{M}\OR%
5037          \equal{\LWR@strresult}{B}\OR%
5038          \equal{\LWR@strresult}{X}%
5039      }%
5040      {%
5041          \allowpars
5042          \LWR@startpars%
5043          \global\booltrue{LWR@tableparcell}%
5044      }%
5045      {}% no pars
```

Print the > contents:

```
5045      \LWR@getexpparray{LWR@colbeforespec}{\theLWR@tablecolspos}%
5046      \global\boolfalse{LWR@intabularmetadata}%
5047  }% not exiting tabular
5048 }{}% in tabular metadata
5049 }%
```

58.15 Midrules

LWR@midrules LWR@midrules is a data array (section 32) of columns containing Y if a midrule should be created for each column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules data array.

```
5050 \newcounter{LWR@midrulecounter}
```

\LWR@clearmidrules Start new midrules. Called at beginning of tabular and also at \\.

Clears all LWR@midrules markers for this line.

```
5051 \newcommand*\LWR@clearmidrules{
```

```

5052 {%
5053 \setcounter{LWR@midrulecounter}{1}%
5054 \whiledo{%
5055 \cntttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolswidth}}}%
5056 }%
5057 {%
5058 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{}%
5059 \addtocounter{LWR@midrulecounter}{1}%
5060 }%
5061 }

```

\LWR@subcmidrule {*width*} {*trim*} {*leftcolumn*} {*rightcolumn*}

Marks LWR@midrules data array elements to be “Y” from left to right columns.

```

5062 \newcommand*{\LWR@subcmidrule}[4]{%
5063 \setcounter{LWR@midrulecounter}{#3}%
5064 \whiledo{\cntttest{\value{LWR@midrulecounter}}{<=}{#4}}{%
5065 {%
5066 \LWR@setexpparray{LWR@midrules}{\theLWR@midrulecounter}{Y}}%
5067 \addtocounter{LWR@midrulecounter}{1}%
5068 }% end of the whiledo
5069 }

```

\LWR@docmidrule [*width*] {*trim*} {*leftcolumn-rightcolumn*}

Marks LWR@midrules array elements to be “Y” from left to right columns.

```

5070 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}m}%
5071 {\LWR@subcmidrule{#1}{#2}{#3}}

```

58.16 Multicolumns

58.16.1 Parsing multicolumns

```
5072 \newcounter{LWR@tablemulticolswidth}
```

Indexes into the multicolumn specification:

```
5073 \newcounter{LWR@tablemulticolspos}
```

\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.

```
5074 \newcommand*\{LWR@printmccoltype}{1}{%
5075 \LWR@traceinfo{lwr@printmccoltype -#1-}%

```

Get one token of the column spec:

```
5076 \StrChar{\#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
5077 \IfStrEq{\LWR@strresult}{l}{\{}%
5078 \IfStrEq{\LWR@strresult}{c}{\{}%
5079 \IfStrEq{\LWR@strresult}{r}{\{}%
5080 \IfStrEq{\LWR@strresult}{p}{\{}%
5081 \IfStrEq{\LWR@strresult}{m}{\{}%
5082 \IfStrEq{\LWR@strresult}{b}{\{}%
5083 \IfStrEq{\LWR@strresult}{P}{\{}%
5084 \IfStrEq{\LWR@strresult}{M}{\{}%
5085 \IfStrEq{\LWR@strresult}{B}{\{}%
5086 \IfStrEq{\LWR@strresult}{S}{\{}%
5087 \IfStrEq{\LWR@strresult}{X}{\{}%
5088 \LWR@traceinfo{lwr@printmccoltype done}%
5089 }
```

\LWR@multicolpartext Print the data with paragraph tags:

```
5090 \newcommand*\{LWR@multicolpartext}{%
5091 \LWR@startpars%
5092 \LWR@multicoltext%
5093 \LWR@stoppars%
5094 }
```

\LWR@multicolother {\langle colspec \rangle} For @, !, >, <, print the next token without paragraph tags:

```
5095 \newcommand*\{LWR@multicolother}{1}{%
5096 \addtocounter{\LWR@tablemulticolspos}{1}%
5097 \StrChar{\#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
5098 \LWR@strresult%
```

A valid column data type was found:

```
5099 \booltrue{LWR@validtablecol}%
5100 }
```

\LWR@multicolskip Nothing to print for this column type.

```
5101 \newcommand*\{LWR@multicolskip}{%
```

A valid column data type was found:

```
5102 \booltrue{LWR@validtablecol}%
5103 }
```

\LWR@printmccoldata {<colspec>} Print the data for any valid column type found.

```
5104 \newcommand*{\LWR@printmccoldata}[1]{%
5105 \LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
5106 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
5107 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
5108 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}%
5109 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}%
5110 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}%
5111 \IfStrEq{\LWR@strresult}{D}%
5112 \addtocounter{LWR@tablemulticolspos}{3}%
5113 \LWR@multicoltext%
5114 }%
5115 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolpartext}%
5116 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolpartext}%
5117 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolpartext}%
5118 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolpartext}%
5119 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolpartext}%
5120 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolpartext}%
5121 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolpartext}%
5122 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolpartext}%
5123 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}%
5124 \IfStrEq{\LWR@strresult}{@}{\LWR@multicolother{#1}}%
5125 \IfStrEq{\LWR@strresult}{!}{\LWR@multicolother{#1}}%
5126 \IfStrEq{\LWR@strresult}{>}{\LWR@multicolother{#1}}%
5127 \IfStrEq{\LWR@strresult}{<}{\LWR@multicolother{#1}}%
```

If an invalid column type:

```
5128 \ifbool{LWR@validtablecol}{}{\LWR@multicoltext}%
```

Tracing:

```
5129 \LWR@traceinfo{lwr@printmccoldata done}%
5130 }
```

\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.

```
5131 \newcommand*\{\LWR@parsemulticolumnalignment}[2]{%
5132 \setcounter{\LWR@tablemulticolspos}{1}%
5133 \StrLen{\#1}[\LWR@strresult]%
5134 \setcounter{\LWR@tablemulticolswidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
5135 \whiledo{%
5136 \not\value{\LWR@tablemulticolspos}>\value{\LWR@tablemulticolswidth}%
5137 }%
5138 {%
```

Execute the assigned print function for each token in the column spec:

```
5139 #2{\#1}%
```

Move to the next token in the column spec:

```
5140 \addtocounter{\LWR@tablemulticolspos}{1}%
5141 }%
5142 }
```

58.16.2 Multicolumn factored code

```
5143 \newcommand{\LWR@multicoltext}{}{}
```

\LWR@domulticolumn [{1: *vpos*} [{2: *#rows*}] {\{3: *numcols*\}} {\{4: *colspec*\}} {\{5: *text*\}}

```
5144 \NewDocumentCommand{\LWR@domulticolumn}{o o m m +m}{%
5145 \LWR@traceinfo{lwr@domulticolumn -#1- -#2- -#3- -#4-} %
```

Remember the text to be inserted, and remember that a valid column type was found:

```
5146 \renewcommand{\LWR@multicoltext}{}{%
5147 #5%
5148 \booltrue{\LWR@validtablecol}%
5149 }%
```

Row processing:

```
5150 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
5151 \LWR@htmltag{td colspan="#3" %
```

```
5152 \IfValueTF{#2}{ % rows?
5153 rowspan="#2" %

5154 \IfValueT{#1}{% vpos?
5155 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
5156 \ifstrequal{#1}{t}{style="vertical-align:top" }{}%
5157 }% vpos?
5158 }{}% rows?

5159 class="td%"
```

Print the column type:

```
5160 \LWR@parsemulticolumnalignment{#4}{\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”.

```
5161 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}}{\theLWR@tablecolspos}}{Y}{rule}{}%
```

Close the class tag’s opening quote:

```
5162 "%
5163 }% end of the opening table data tag
5164 \global\boolfalse{\LWR@intabularmetadata}%
5165 \LWR@parsemulticolumnalignment{#4}{\LWR@printmccoldata}%
5166 }
```

58.16.3 Multicolumn

```
\LWR@htmlmulticolumn {⟨numcols⟩} {⟨alignment⟩} {⟨text⟩}
```

```
5167 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
5168 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5169 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn tag:

```
5170 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{#2}{#3}%
```

Move to the next L^AT_EX column:

```
5171 \addtocounter{LWR@tablecolspos}{#1}%
5172 \addtocounter{LWR@tablecolspos}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
5173 \booltrue{LWR@skipatbang}%
5174 }
```

58.16.4 Longtable captions

longtable captions use \multicolumn.

Bool LWR@starredlongtable Per the caption pacakge, step the counter if longtable*.

```
5175 \newbool{LWR@starredlongtable}
5176 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5177 \providetoggle*{\LTcaptype}{table}
```

```
\LWR@longtabledatacaptiontag * [(toc entry)] {(caption)}

5178 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5179 {%
```

Remember the latest name for \nameref:

```
5180 \IfValueTF{#2}{% optional given?
5181   \ifthenelse{\equal{#2}{}}% optional empty?
5182     {\LWR@setlatestname{#3}}% empty
5183     {\LWR@setlatestname{#2}}% given and non-empty
5184 }% optional given
5185 {\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:

```
5186 \LWR@tabularhtmlcolumns{1}{\theLWR@tabletotalcols}
```

Create the multicolumn tag:

```
5187 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{P}{%
5188 \IfBooleanTF{#1}{ star? }
```

Star version, show a caption but do not make a LOT entry:

```
5189 {%
5190   yes star
5191   \LWR@htmlblocktag{figcaption}%
5192   #3%
5193   \LWR@htmlblocktag{/figcaption}%
5194 }%
5194 {%
5194   No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```
5195 \ifbool{\LWR@starredlongtable}{%
5196 {%
5197 \ifthenelse{\equal{#2}{}}{%
5198 {}%
5199 {%
5200   \refstepcounter{LTcaptive}%
5201   \protected@edef\@currentlabel{%
5202     \csuse{p@\LTcaptive}\csuse{the\LTcaptive}}%
5203   }%
5204 }{}}
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
5205 \LWR@htmlblocktag{figcaption}%
5206 \csuse{fnum@\LTcaptive}\CaptionSeparator#3%
5207 \LWR@htmlblocktag{/figcaption}%
```

See if an optional caption was given:

```
5208 \ifthenelse{\equal{#2}{}}{%
5208   TOC entry empty}
```

if the optional caption was given, but empty, do not form a TOC entry

```
5209 {}%
```

If the optional caption was given, but might only be []:

```
5210 {%
5211   TOC entry not empty
5211   \IfNoValueTF{#2}{%
5211     No TOC entry? }
```

The optional caption is []:

```

5212      {%
5213          No TOC entry
5214          \addcontentsline%
5215          {\csuse{ext@\LTcaptype}}%
5216          {\LTcaptype}%
5217          {%
5218              \protect\newline%
5219              {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
5220              {\ignorespaces #3\protect\relax}%
5221          }%
5222      }% end of No TOC entry

```

The optional caption has text enclosed:

```

5222      {%
5223          Yes TOC entry
5224          \addcontentsline%
5225          {\csuse{ext@\LTcaptype}}%
5226          {\LTcaptype}%
5227          {%
5228              \protect\newline%
5229              {\csuse{p@\LTcaptype}\csuse{the\LTcaptype}}%
5230              {\ignorespaces #2\protect\relax}%
5231          }%
5232      }% end of yes TOC entry
5233  }% end of TOC entry not empty
5234 }% end of no star

```

Skip any trailing @ or ! columns for this cell:

```

5234 \booltrue{LWR@skipatbang}%
5235 }% end of \LWR@domulticolumn
5236
5237 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
5238 \addtocounter{LWR@tablecolspos}{-1}
5239
5240 }

```

58.16.5 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.

```
5241 \newcounter{LWR@tabhtmlcolindex}
5242 \newcounter{LWR@tabhtmlcolend}
5243 \newcounter{LWR@tabhtmlcoltotal}
```

`\LWR@tabularhtmlcolumns {<starting LTEX column>} {<number LTEX columns>}`

Compute the total number of HTML columns being spanned, considering the starting L^TE_X table column and the number of L^TE_X 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.

```
5244 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the L^TE_X span, to which additional @ and ! columns may be added:

```
5245 \setcounter{LWR@tabhtmlcolindex}{#1}%
5246 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5247 \setcounter{LWR@tabhtmlcolend}{#1}%
5248 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

Walk across the L^TE_X columns looking for @ and ! columns:

```
5249 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
```

Temporarily define a macro equal to the @ specification for this column:

```
5250 \edef\LWR@atbangspec{\LWR@getexarray{LWR@colatspec}{\theLWR@tabhtmlcolindex}}%
```

If the @ specification is not empty, add to the count:

```
5251 \ifdefempty{\LWR@atbangspec}{}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
5252 \edef\LWR@atbangspec{\LWR@getexarray{LWR@colbangspec}{\theLWR@tabhtmlcolindex}}%
5253 \ifdefempty{\LWR@atbangspec}{}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Move to the next L^TE_X column:

```
5254 \addtocounter{LWR@tabhtmlcolindex}{1}%
5255 }%
```

If at the left-most column, also skip the leftmost @ and ! cells:

```

5256 \ifthenelse{\value{LWR@tablecolspos}=1}{%
5257   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{leftedge}}%
5258   \ifdefempty{\LWR@atbangspec}{}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5259   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{leftedge}}%
5260   \ifdefempty{\LWR@atbangspec}{}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5261 }{}%
5262 }

5263 \end{warpHTML}

```

58.17 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 168 for the HTML versions.

```

for HTML & PRINT: 5264 \begin{warpall}

\multicolumnrow {\langle 1:cols\rangle} {\langle 2:align\rangle} [\langle 3:vpos\rangle] {\langle 4:numrows\rangle} [\langle 5:bigstruts\rangle] {\langle 6:width\rangle} [\langle 7:fixup\rangle]
{\langle 8:text\rangle}

```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:
<https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex>

```

5265 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}{m m 0{c} m 0{0} m 0{opt} +m}{%
5266 \multicolumn{\#1}{\#2}{\@xmultirow[\#3]{\#4}{\#5}{\#6}{\#7}{\#8}}%
5267 }
5268
5269 \newcommand*{\multicolumnrow}{\LWR@origmulticolumnrow}

5270 \end{warpall}

```

58.18 Utility macros inside a table

```

for HTML output: 5271 \begin{warpHTML}

5272 \newcommand*{\LWR@donothing}{}
5273 \newcommand*{\LWR@domidrule}{\booltrue{LWR@doinghline}}
5274 \newcommand*{\LWR@dotbrule}{\booltrue{LWR@doingtbrule}}

5275 \end{warpHTML}

```

58.19 Special-case tabular markers

for HTML & PRINT: 5276 \begin{warpall}

\TabularMacro Place this just before inserting a custom macro in a table data cell. Doing so tells lwarf not to automatically start a new HTML table data cell yet. See section 7.7.

5277 \let\TabularMacro\relax

5278 \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, lwarf's emulation of the tabular does not automatically resume when the containing environment 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
}
\EndDefiningTabulars
```

for HTML output: 5279 \begin{warpHTML}

```
5280 \newcommand*\{\ResumeTabular\}{%
 5281 \global\boolfalse{LWR@exittingtabular}%
 5282 \LWR@getmynexttoken%
 5283 }
```

5284 \end{warpHTML}

for PRINT output: 5285 \begin{warpprint}

```
5286 \newcommand*\{\ResumeTabular\}{}
5287 \end{warpprint}
```

58.20 Checking for a new table cell

for HTML output: 5288 \begin{warpHTML}

\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:

```
5289 \newbool{\LWR@exittingtabular}
5290
5291 \newcommand*\LWR@tabledatacolumntag{%
5292 {%
5293 \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:

```
5294 \let\mynext\LWR@tabledatasinglecolumntag{%
```

If exiting the tabular:

```
5295 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\end}}{%
5296 {\booltrue{\LWR@exittingtabular}}{}}
```

longtable can have a caption in a cell

```
5297 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\caption}}{%
5298 {\let\mynext\LWR@donothing}{}}
```

Look for other things which would not start a table cell:

```
5299 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumn}}{%
5300 {\let\mynext\LWR@donothing}{}}
5301 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multirow}}{%
5302 {\let\mynext\LWR@donothing}{}}
5303 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\multicolumnrow}}{%
5304 {\let\mynext\LWR@donothing}{}}
```

if come to an \mrowcell, this is a cell to be skipped over

```
5305 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\mrowcell}}{%
5306 {\let\mynext\LWR@donothing}{}}
5307 %
5308 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\TabularMacro}}{%
5309 {\let\mynext\LWR@donothing}{}}
5310 %
```

```

5311 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\hline}}{%
5312 {\let\mynext\LWR@donothing}{}%}
5313 %
5314 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\toprule}}{%
5315 {\let\mynext\LWR@donothing}{}%}
5316 %
5317 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\midrule}}{%
5318 {\let\mynext\LWR@donothing}{}%}
5319 %
5320 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cmidrule}}{%
5321 {\let\mynext\LWR@donothing}{}%}
5322 %
5323 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\specialrule}}{%
5324 {\let\mynext\LWR@donothing}{}%}
5325 %
5326 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\cline}}{%
5327 {\let\mynext\LWR@donothing}{}%}
5328 %
5329 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\bottomrule}}{%
5330 {\let\mynext\LWR@donothing}{}%}
5331 %
5332 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpprntonly}}{%
5333 {\let\mynext\LWR@donothing}{}%}
5334 %
5335 \ifthenelse{\isequivalentto{\LWR@mynexttoken}{\warpHTMLonly}}{%
5336 {\let\mynext\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:

```

5337 \mynext%
5338 }

5339 \end{warpHTML}

```

58.21 \mrowcell

\mrowcell The user must insert `\mrowcell` into any multirow cells which must be skipped. This command has no action during print output.

for HTML & PRINT:

```

5340 \begin{warpall}
5341 \newcommand*{\mrowcell}{}%
5342 \end{warpall}

```

58.22 New \tabular definition

for HTML output: 5343 \begin{warpHTML}

These are default definitions in case booktabs is not loaded, and are not expected to be used, but must exist as placeholders.

```
5344 \newcommand*{\LWR@origtoprule}[1]{\hline}
5345 \newcommand*{\LWR@origmidrule}[1]{\hline}
5346 \LetLtxMacro{\LWR@origcmidrule}{\cline}
5347 \newcommand*{\LWR@origbottomrule}[1]{\hline}
5348 \newcommand*{\LWR@origaddlinespace}[1][]{}
5349 \newcommand*{\LWR@origmorecmidrules}(){}
5350 \newcommand*{\LWR@origspecialrule}[3]{\hline}
```

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.

```
5351 \StartDefiningTabulars
5352
5353 \newenvironment*{\LWR@tabular}[2] []
5354 {%
5355 \LWR@traceinfo{\LWR@tabular started}%
5356 \addtocounter{\LWR@tabulardepth}{1}%

```

Not yet started a table row:

```
5357 \global\boolfalse{\LWR@startedrow}%
```

Not yet doing an hline:

```
5358 \global\boolfalse{\LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
5359 \global\boolfalse{\LWR@doingtbrule}%
```

For babel-french:

```
5360 \LWR@FBcancel%
```

Have not yet found the end of tabular command:

```
5361 \boolfalse{\LWR@exittingtabular}%
```

Create the table tag:

```
5362 \global\booltrue{LWR@intabularmetadata}%
5363 \LWR@forcenewpage
5364 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
5365 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecols which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
5366 \LWR@stoppars%
```

Track column # for setting text-align:

```
5367 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5368 \LWR@clearmidrules%
```

\\" becomes a macro to end the table row:

```
5369 \LetLtxMacro{\\"}{\LWR@tabularendofline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with \LWR@getmynexttoken to see if the next token might create a new data cell:

```
5370 \LWR@traceinfo{LWR@tabular: redefining macros}%
5371 \renewcommand*{\hline}{\LWR@domidrule\LWR@getmynexttoken}%
5372
5373 \RenewDocumentCommand{\cline}{m}%
5374 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
5375
5376 \DeclareDocumentCommand{\toprule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5377
5378 \DeclareDocumentCommand{\midrule}{o}{\LWR@domidrule\LWR@getmynexttoken}%
5379
5380 \DeclareDocumentCommand{\cmidrule}{o d() m}%
5381 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
5382
5383 \DeclareDocumentCommand{\bottomrule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5384 \DeclareDocumentCommand{\addlinespace}{o}{}%
5385 \DeclareDocumentCommand{\morecmidrules}{}{%
5386 \DeclareDocumentCommand{\specialrule}{m m m}{\LWR@domidrule\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.

```
5387 \renewcommand{\multicolumn}{\LWR@htmlemulticolumn}%
5388 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
5389 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}
```

Reset for new processing:

```
5390 \global\boolfalse{\LWR@tableparcell}%
5391 \global\boolfalse{\LWR@skippingmrowcell}%
5392 \global\boolfalse{\LWR@skipatbang}
```

Set & for its special meaning inside the tabular:

```
5393 \StartDefiningTabulars%
5394 \protected\gdef&{\LWR@tabularampersand}%
```

Look ahead for a possible table data cell:

```
5395 \LWR@traceinfo{\LWR@tabular: about to \LWR@getmynexttoken}%
5396 \LWR@getmynexttoken%
5397 }%
```

Ending the environment:

```
5398 {%
5399 \LWR@traceinfo{\LWR@tabular ending}%
5400 \LWR@closetabledatacell%
5401 \LWR@htmleblocktag{/tr}%
5402 \LWR@htmleblocktag{/table}%
5403 \global\boolfalse{\LWR@intabularmetadata}}
```

Unnest one level of tabular:

```
5404 \addtocounter{\LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
5405 \protected\gdef&{\LWR@origampmacro}%
5406 \EndDefiningTabulars%
5407 \LWR@traceinfo{\LWR@tabular finished ending}%
5408 }%
5409
5410 \EndDefiningTabulars
```

```
5411 \end{warpHTML}
```

58.23 Array

```
Pkg array
array is also automatically loaded by siunitx.
```

59 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 8 shows the data structures related to cross-referencing.

for HTML output: 5412 \begin{warpHTML}

59.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

5413 \newcommand*\{@currentlabelname}{}%

\LWR@stripperiod {\langle text \rangle} [.\.]

Removes a trailing period.

5414 \def\LWR@stripperiod#1.\ltx@empty#2@nil{#1}%

\LWR@setlatestname {\langle object name \rangle}

Removes \label, strips any final period, and remembers the result.

5415 \newcommand*\LWR@setlatestname[1]{%

Remove \label and other commands from the name, the strip any final period. See zref-titleref and gettitlestring.

5416 \GetTitleStringExpand{#1}%

Table 8: Cross-referencing data structures

Original L^AT_EX:	(print and HTML)
\refstepcounter: Steps the counter 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 l warp:	(HTML only)
\label: Adds HTML tags (section 59.3), plus \splabel data (section 59.2):	
zLWR@name: The section name for this label.	
zLWR@htmlfilenumber: The filenumber or name for this label.	
zLWR@teximagedepth: The teximagedepth for this label.	
zLWR@teximagenumber: The teximagenumber for this label.	
\nameref: Emulated from hyperref for l warp. See section 59.4.	
\ref and \nameref: Adds HTML tags. See section 59.4.	
Added by amsmath:	(print and HTML)
\label: Execution is delayed until the math environment is completed.	
\ltx@label: L ^A T _E X \label, (HTML: patched by l warp,) 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 45.4 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 l warp:	(HTML only)
\cref, etc.: Modified for l warp. See section 69.	
\label inside math: See section 63.4.1.	
Footnotes: See \noteentry in section 45.4.	

```

5417 \edef@\currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5418 \edef@\currentlabelname{%
5419 \expandafter\LWR@stripperiod@\currentlabelname%
5420 \ltx@empty.\ltx@empty@\nil%
5421 }%
5422 }

```

59.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called special:

```
5423 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5424 \zref@newprop{zLWR@name}{\currentlabelname}
```

Define a new property which has either a filename or a file number:

```

5425 \zref@newprop{zLWR@htmlfilename}{%
5426 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilename}%
5427 }%

```

Additional properties for lateximages:

```

5428 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5429 \zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}

```

`zLWR@htmlfilename` property holds the file number or name

Add a `LWR@htmlfilename` property, and `lateximage` properties to `special`:

```

5430 \zref@addprop{special}{zLWR@name}
5431 \zref@addprop{special}{zLWR@htmlfilename}
5432 \zref@addprop{special}{zLWR@lateximagedepth}
5433 \zref@addprop{special}{zLWR@lateximagenumber}

```

Returns the selected field:

```

5434 \newcommand*{\LWR@spref}[2]{%
5435 \zref@extractdefault{#1}{#2}{??}}

```

\LWR@nameref {*label*} Returns the section name for this label:

```
5436 \newcommand*\{\LWR@nameref\}[1]{%
5437 \LWR@sref{\#1}{zLWR@name}%
5438 }
```

\LWR@htmlfileref {*label*} Returns the file number for this label:

```
5439 \newcommand*\{\LWR@htmlfileref\}[1]{%
5440 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5441 \LWR@sref{\#1}{zLWR@htmlfilenumber}%
5442 }
```

\LWR@lateximagedepthref {*label*} Returns the lateximagedepth for this label:

```
5443 \newcommand*\{\LWR@lateximagedepthref\}[1]{%
5444 \LWR@sref{\#1}{zLWR@lateximagedepth}%
5445 }
```

\LWR@lateximagenumberref {*label*} Returns the lateximagenumber for this label:

```
5446 \newcommand*\{\LWR@lateximagenumberref\}[1]{%
5447 \LWR@sref{\#1}{zLWR@lateximagenumber}%
5448 }
```

\LWR@splabel {*label*} Sanitize the name and then creates the label:

```
5449 \newcommand*\{\LWR@splabel\}[1]{%
5450 \LWR@setlatestname{\@currentlabelname}%
5451 \zref@labelbylist{\#1}{special}}
```

59.3 Labels

\LWR@subsublabel {*label*} Creates an HTML id tag.

```
5452 \newcommand*\{\LWR@subsublabel\}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5453 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}>}{0}{}%
5454 {}%
5455 % not lateximage
```

If not doing a `\teximage`, create an HTML ID tag: (To be factored...)

```

5456 \ifbool{LWR@doingstartpars}%
5457 {%
5458   pars allowed
5459   \ifbool{LWR@doingapar}%
5460     {%
5461       par started
5462       \LWR@htmltag{a id="#1"\{}\LWR@htmltag{/a}\%
5463     }%
5464     par started
5465     \LWR@startpars%
5466   }%
5467   par not started
5468   \LWR@stoppars%
5469   \LWR@htmltag{a id="#1"\{}\LWR@htmltag{/a}\%
5470 }%
5471 pars not allowed
5472 }%
5473 not \teximage
5474 }

```

`\LWR@newlabel {<label>} [<type>]`

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular \LaTeX `\label`.

`clevereref` later encases this to add its own cross-referencing.

The optional `<type>` is per the `ntheorem` package, and is ignored.

```

5473 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5474   \LWR@traceinfo{LWR@newlabel: starting}%
5475   \LWR@traceinfo{LWR@newlabel: !#1!}%
5476   \esphack%

```

Create a traditional \LaTeX label, as modified by `cleveref`:

```
5477 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, `LWR@htmlfilename`, `LWR@teximagedepth`, and `LWR@teximagenumber`:

```

5478 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5479 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5480 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilename is \theLWR@htmlfilename}%
5481 \LWR@splabel{#1}%
5482 \LWR@subsublabel{#1}%
5483 \esphack%
5484 \LWR@traceinfo{LWR@newlabel: done}%
5485 }

```

59.4 References

\LWR@startref {*label*} (Common code for \ref and \nameref.)

Open an HTML tag reference to a filename, # character, and a label.

```
5486 \newcommand*\LWR@startref[1]
5487 {%
5488 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
5489 \LWR@traceinfo{\LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
5490 \LWR@htmlltag{a href="%"
5491 \LWR@traceinfo{\LWR@startref B}%
5492 \LWR@htmlrefsectionfilename{#1}%
5493 \LWR@traceinfo{\LWR@startref C}%
5494 %#"
```

Create the destination id:

See if \LWR@lateximagedepth is unknown:

```
5495 \LWR@traceinfo{\LWR@startref D: !#1!}%
5496 \ifthenelse{\equal{\LWR@lidref}{??}}{%
```

“??” if \LWR@lateximagedepth is unknown, so create a link with an unknown destination:

```
5497 {%
5498     \LWR@traceinfo{\LWR@startref D0: ??}%
5499     ??%
5500 }%
```

If \LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```
5501 {%
5502     \LWR@traceinfo{\LWR@startref D1: \LWR@lidref}%
5503     \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}{%
5504         \LWR@traceinfo{\LWR@startref D2: \LWR@lidref}%
5505         lateximage\LWR@lateximagenumberref{#1}%
5506     }{%
5507     }%
5508     \LWR@traceinfo{\LWR@startref D3}%
5509     #1%
5511 }%
```

```
5512 }%
5513 \LWR@traceinfo{LWR@startref E}%
```

Closing quote:

```
5514 "{}"%
5515 \LWR@traceinfo{LWR@startref F}%
5516 }
```

\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)”.

```
5517 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5518 \LWR@traceinfo{LWR@subnewref #1 #2}%
5519 \LWR@startref{#1}%
5520 \LWR@origref{#2}%
5521 \LWR@htmltag{/a}%
5522 }
```

\ref * {<label>} \ref is \let to \LWR@newref

\LWR@newref * {<label>} Create an internal document reference link, or without a link if starred per hyperref.

```
5523 \NewDocumentCommand{\LWR@newref}{s m}{%
5524 \LWR@traceinfo{LWR@newref #2}%
5525 \IfBooleanTF{#1}%
5526 {\LWR@origref{#2}}%
5527 {\LWR@subnewref{#2}{#2}}%
5528 }
```

\pagerefPageFor Text for starred page references.

```
5529 \newcommand*{\pagerefPageFor}{see }
```

\pageref * {<label>} Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
5530 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5531 \IfBooleanTF{#1}%
5532 {(\pagerefPageFor\LWR@origref{#2})}%
5533 {(\cpageref{#2})}%
5534 }
```

```
\nameref {<label>}

5535 \newcommand*{\nameref}[1]{%
5536 \LWR@traceinfo{\nameref A}%
5537 \LWR@startref{#1}%
5538 \LWR@traceinfo{\nameref B}%
5539 \LWR@nameref{#1}%
5540 \LWR@traceinfo{\nameref C}%
5541 \LWR@htmltag{/a}%
5542 \LWR@traceinfo{\nameref D}%
5543 }
```

\Nameref {<label>} In print, adds the page number. In HTML, does not.

```
5544 \let\Nameref\nameref
```

59.5 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.

```
5545 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
5546 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

```
\LWR@subhyperref {<URL>} {<text>}
```

```
5547 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5548 \LWR@htmltag{a href="#1" target="\_blank"\LWR@newline}\#2\LWR@htmltag{/a}%
5549 \LWR@ensuredoingapar%
5550 }
```

```
\LWR@subhyperrefclass {<URL>} {<text>} {<htmlclass>}
```

```

5551 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5552 \LWR@htmltag{a href="#1"%
5553 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5554 \LWR@ensuredoingapar%
5555 }

```

`\href [⟨options⟩] {⟨URL⟩} {⟨text⟩}`

Create a link with accompanying text:

```

5556 \NewDocumentCommand{\LWR@hrefb}{O{} m +m}{%
5557 \LWR@subhyperref{#2}{#3}%
5558 \endgroup%
5559 \LWR@ensuredoingapar%
5560 }
5561
5562 \newcommand{\href}{%
5563 \LWR@ensuredoingapar%
5564 \begingroup%
5565 \catcode`\_=12
5566 \LWR@hrefb%
5567 }

```

`\nolinkurl {⟨URL⟩}`

Print the name of the link without creating the link:

```

5568 \newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
5569
5570 \newcommand{\nolinkurl}{%
5571 \LWR@ensuredoingapar%
5572 \begingroup\catcode`\_=12
5573 \LWR@nolinkurlb%
5574 }

```

`\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`.

```

5575 \newcommand*{\LWR@urlb}[1]{%
5576 \href{#1}{#1}%
5577 \endgroup%
5578 \LWR@ensuredoingapar%
5579 }
5580
5581 \newcommand{\LWR@urla}{%
5582 \LWR@ensuredoingapar%

```

```
5583 \begingroup\catcode`\_=12
5584 \LWR@urlb%
5585 }
5586
5587 \let\url\LWR@urla

\LWR@subinlineimage  [{\it alttag}] {\it class} {\it filename} {\it extension} {\it style}

5588 \newcommand*{\LWR@subinlineimage}[5] []{%
5589 \ifthenelse{\equal{#1}{}}{%
5590 {\LWR@htmltag{img src="#3.\#4" alt="#3" style="#5" class="#2"[]}}%
5591 {\LWR@htmltag{img src="#3.\#4" alt="#1" style="#5" class="#2"[]}}%
5592 }{%
5593 \end{warpHTML}
```

Table 9: 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 `\TeX` file identifier for the output file.

`\LWR@have<type>`: A boolean rememebering 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><name>` “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.

60 Floats

Floats are supported, although partially through emulation.

Table 9 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”.

60.1 Float captions

for HTML output: 5594 \begin{warpHTML}

\LWR@floatbegin {*type*} [*placement*]

Begins a \newfloat environment.

```
5595 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5596   ifthenelse{\boolean{FormatWordProcessor}}\AND\boolean{HTMLMarkFloats}}{%
5597   %
5598   === #1 begin
5599   %
5600 }{%
5601 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5602 \addtocounter{\LWR@thisfloat}{1}%
5603 \booltrue{\LWR@freezethisfloat}%
5604 \begingroup
```

Settings while inside the environment:

```
5605 \LWR@origraggedright
```

Open an HTML figure tag:

```
5606 \LWR@htmntag{figure id="autofloat-\arabic{\LWR@thisfloat}" class="#1"}%
5607 \renewcommand*{\@capttype}{#1}
5608 \caption@settype{#1}
5609 \LWR@startpars
5610 }
```

\@float Support packages which create floats directly.
\@dblfloat
5611 \let\@float\LWR@floatbegin
5612 \let\@dblfloat\LWR@floatbegin

\LWR@floatend Ends a \newfloat environment.

```
5613 \newcommand*{\LWR@floatend}{%
5614 \LWR@stoppars%
5615 \LWR@htmlelementend{figure}}%
```

```

5616 \endgroup%
5617 \boolfalse{LWR@freezethisfloat}%
5618 \LWR@startpars%
5619 \ifthenelse{\boolean{FormatWordProcessor}}{\AND\boolean{HTMLMarkFloats}}{%
5620
5621 === end
5622
5623 }{}%
5624 }

```

\end@float Support packages which create floats directly.
\end@dblfloat

```

5625 \let\end@float\LWR@floatend
5626 \let\end@dblfloat\LWR@floatend

```

Ctr LWR@thisfloat 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.

```
5627 \newcounter{LWR@thisfloat}
```

Bool LWR@freezethisfloat Prevents multiple increments of \LWR@thisfloat inside a float.

```

5628 \newbool{LWR@freezethisfloat}
5629 \boolfalse{LWR@freezethisfloat}

```

\LWR@maybeincthisfloat

```

5630 \newcommand*{\LWR@maybeincthisfloat}{%
5631 \ifbool{LWR@freezethisfloat}{}{\addtocounter{LWR@thisfloat}{1}}%
5632 }

```

\@captype Remembers which float type is in use.

```
5633 \newcommand*{\@captype}{}%
```

60.1.1 Caption inside a float environment

\CaptionSeparator How to separate the float number and the caption text.

```
5634 \AtBeginDocument{\providecommand*{\CaptionSeparator}{:~}}
```

\@makecaption {\langle name and num \rangle} {\langle text \rangle}

Prints the float type and number, the caption separator, and the caption text.

```
5635 \AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}
```

60.1.2 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 `autofloat` `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` `autofloat <id>` at the start of the float itself in the `HTML` file. The `autopage` and `autofloat` 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@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.
- Ctr `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
5636 \newcounter{LWR@nextautofloat}
5637 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat {<autopage>} {<autofloat>}`

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autofloat` are remembered for `\l@figure` to use when creating the `HTML` links.

```
5638 \newcommand*{\LWRsetnextfloat}[2]{%
5639 \setcounter{LWR@nextautopage}{#1}%
5640 \setcounter{LWR@nextautofloat}{#2}%
5641 }
```

- Ctr `LWR@latestautopage` Updated each time a new `HTML` file is begun. `\LWRsetnextfloat` is written with this and the `autofloat` by the modified `\addcontentsline` just before each float's entry.

```
5642 \newcounter{LWR@latestautopage}
5643 \setcounter{LWR@latestautopage}{1}

5644 \let\LWR@origcaption@begin\caption@begin
5645 \let\LWR@origcaption@end\caption@end
5646 \let\LWR@orig@@par\@@par
```

`\LWR@caption@begin` Low-level patches to create `HTML` tags for captions.

```
5647 \newcommand{\LWR@caption@begin}{%
5648 %
5649 \LWR@traceinfo{LWR@caption@begin}%
}
```

Keep `par` and `minipage` changes local:

```
5650 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5651 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a `minipage` or `\parbox` inside the caption:

```
5652 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{}{%
5653 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{{\#5}}%
```

Enclose the original caption code inside an HTML tag:

```
5654 \LWR@htmlblocktag{figcaption}%
5655 \LWR@origcaption@begin%
5656 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5657 \newcommand{\LWR@caption@end}%
5658 {%
5659 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5660 \LWR@origvspace*\{\baselineskip}%
```

Closing tag:

```
5661 \LWR@htmlblocktag{/figcaption}%
5662 \endgroup%
5663 % \leavevmode% avoid bad space factor (0) error
5664 \LWR@traceinfo{\LWR@caption@end: done}%
5665 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.

```
\caption@end
5666 \AtBeginDocument{%
5667 \let\caption@begin\LWR@caption@begin
5668 \let\caption@end\LWR@caption@end
5669 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
5670 \let\LWR@origcaptionlistentry\captionlistentry
5671
5672 \renewcommand*{\captionlistentry}{%
```

```

5673 \LWR@maybeincthisfloat%
5674 \LWR@ensuredoingapar%
5675 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"}\LWR@htmlltag{/a}%
5676 \LWR@origcaptionlistentry%
5677 }
5678
5679 \def\LWR@LTcaptionlistentry{%
5680 \LWR@ensuredoingapar%
5681 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"}\LWR@htmlltag{/a}%
5682 \bgroup
5683 \ifstar{\egroup\LWR@LT@captionlistentry}{%
5684 \egroup\LWR@LT@captionlistentry}%
5685 \def\LWR@LT@captionlistentry#1{%
5686 \caption@listentry\firsofttwo[\LTcaptype]{#1}}%

```

- \addcontentsline** Patched to write the autopage and autofloat 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.

```

5687 \let\LWR@origaddcontentsline\addcontentsline
5688
5689 \renewcommand*\addcontentsline[3]{%
5690 \ifthenelse{\equal{#1}{toc}}{}{%
5691 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{%
5692 \addtocontents{\nameuse{ext@#2}}{%
5693 \protect\LWRsetnextfloat{%
5694 {\arabic{\LWR@latestautopage}}{%
5695 {\arabic{\LWR@thisfloat}}{%
5696 }% addtocontents
5697 }% not toc
5698 \LWR@origaddcontentsline{#1}{#2}{#3}%
5699 }

```

- \captionof** Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5700 \AtBeginDocument{
5701 \let\LWR@origcaptionof\captionof
5702
5703 \renewcommand*\captionof{%
5704 \LWR@maybeincthisfloat%
5705 \LWR@stopars
5706 \LWR@htmlltag{a id="autofloat-\arabic{\LWR@thisfloat}"}\LWR@htmlltag{/a}%
5707 \LWR@origcaptionof%
5708 }
5709 }

5710 \end{warpHTML}

```

61 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 L^AT_EX infrastructure is used for toc, along with some patches to generate HTML output.

for HTML output: 5711 \begin{warpHTML}

61.1 Reading and printing the TOC

\LWR@myshorttoc {\langle toc/lof/lot\rangle}

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 L^AT_EX, the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sidetoc.

5712 \newcommand*{\LWR@myshorttoc}[1]{
5713 \LWR@ensuredoingapar

Only if the file exists:

5714 \IfFileExists{\jobname.\#1}{

 Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.

 **disabled** For pdflatex, also change to latin1 encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it utf8.

5715 \begingroup
5716 % \ifxetexorluatex%
5717 % \else
5718 % \inputencoding{latin1}% currently disabled
5719 % \fi

```
5720 \makeatletter
```

Read in the TOC file:

```
5721 \@input{\jobname.\#1}
5722 % \makeatother
5723 \endgroup
5724 }%
5725 {%
5726 }
```

```
\LWR@subtableofcontents {\langle toc/lof/lot\rangle} {\langle sectionstarname\rangle}
```

Places a TOC/LOF/LOT at the current position.

```
5727 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
5728 \@ifundefined{chapter}
5729 {\LWR@closeprevious{\LWR@depthsection}}
5730 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
5731 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
5732 \@ifundefined{chapter}{\section*{\#2}}{\chapter*{\#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
5733 \LWR@htmlelementclass{nav}{\#1}
```

Create the actual list:

```
5734 \LWR@myshorttoc{\#1}
```

Close the nav:

```
5735 \LWR@htmlelementclassend{nav}{\#1}
5736 }
```

Patch \starttoc to encapsulate the TOC inside HTML tags:

```
5737 \let\LWR@orig@starttoc\@starttoc
```

```

5738
5739 \renewcommand{\@starttoc}[1]{
5740 \LWR@htmlelementclass{nav}{#1}
5741 \LWR@orig@starttoc{#1}
5742 \LWR@htmlelementclassend{nav}{#1}
5743 }
```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```

5744 \let\LWR@origtableofcontents\tableofcontents
5745 \let\LWR@origlistoffigures\listoffigures
5746 \let\LWR@origlistoftables\listoftables
5747
5748 \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:

```
5749 \ifbool{FormatWordProcessor}{}{
```

Copy the `.toc` file to `.sidetoc` for printing the `sideroc`. The original `.toc` file is renewed when `\tableofcontents` is finished.

```

5750     \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
5751     \LWR@printpendingfootnotes
5752     \LWR@origtableofcontents
5753 }
5754 }
5755 \renewcommand*{\listoffigures}{%
5756 \ifbool{FormatWordProcessor}{}{%
5757     \LWR@printpendingfootnotes
5758     \LWR@origlistoffigures
5759 }
5760 }
5761
5762 \renewcommand*{\listoftables}{%
5763 \ifbool{FormatWordProcessor}{}{%
5764     \LWR@printpendingfootnotes
5765     \LWR@origlistoftables
5766 }
5767 }
```

61.2 High-level TOC commands

```
\listof {<type>} {<title>}
```

Emulate the `\listof` command from the `float` package (section 124). Used to create lists of custom float types. Also used to redefine the standard L^AT_EX `\listoffigures` and `\listoftables` commands.

```
5768 \NewDocumentCommand{\listof}{m +m}{%
5769 \LWR@subtableofcontents{\@nameuse{ext@\#1}}{#2}
5770 \expandafter\newwrite\csname tf@\csname ext@\#1\endcsname\endcsname
5771 \immediate\openout \csname tf@\csname ext@\#1\endcsname\endcsname
5772   \jobname.\csname ext@\#1\relax
5773 }
```

61.3 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.

```
5774 \end{warpHTML}
```

for HTML & PRINT: 5775 `\begin{warpall}`

Ctr SideTOCDepth Controls how deep the side-TOC gets. Use a standard L^AT_EX section level similar to `tocdepth`.

```
5776 \newcounter{SideTOCDepth}
5777 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sidetoc.

```
5778 \newcommand{\sidetocname}{Contents}
5779 \end{warpall}
```

for HTML output: 5780 `\begin{warpHTML}`

\LWR@sidetoc Creates the actual side-TOC.

```
5781 \newcommand*\LWR@sidetoc{%
5782 \LWR@forcenewpage
5783 \LWR@stoppars
5784 }
```

The entire sidetoc is placed into a nav of class sidetoc.

```
5785 \LWR@htmlelementclass{nav}{sidetoc}
5786
5787 \setcounter{tocdepth}{\value{SideTOCDepth}}
5788 }
```

The title is placed into a <div> of class sidetoctitle, and may contain paragraphs.

```
5789 \begin{BlockClass}{sidetoctitle}
5790 \sidetocname
5791 \end{BlockClass}
```

The table of contents is placed into a <div> of class sidetoccontents.

```
5792 \begin{BlockClass}{sidetoccontents}
5793 \LinkHome
5794
5795 \LWR@myshorttoc{sidetoc}
5796 \end{BlockClass}
5797 \LWR@htmlelementclassend{nav}{sidetoc}
5798 }
```

61.4 Low-level TOC line formatting

\numberline {⟨number⟩}

(Called from each line in the .aux, .lof files.)

Record this section number for further use:

```
5799 \renewcommand*\numberline[1]{%
5800 \LWR@sectionnumber{\#1}\quad%
5801 }
```

\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.

- #1 is depth
- #2 is section, subsection, etc.
- #3 the text of the caption
- #4 page number

```
5802 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
5803 \ifthenelse{\cnttest{#1}{<=}{\value{tocdepth}}}{%
5804     \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
5805     \LWR@subhyperrefclass{%
5806         \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
5807         \LWR@stoppars%
5808 }
5809 {}
5810 }
```

Ctr lofdepth TOC depth for figures.

```
5811 \newcounter{lofdepth}
5812 \setcounter{lofdepth}{1}
```

Ctr lotdepth TOC depth for tables.

```
5813 \newcounter{lotdepth}
5814 \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

```
5815 \newcommand{\hypertocfloat}[5]{%
5816 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
5817 \@ifundefined{c@#3depth}{%
5818 \newcounter{#3depth}%
5819 \setcounter{#3depth}{1}%
5820 }{}%
```

Respond to `lofdepth`, etc.:

```
5821 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
5822 \ifthenelse{\cnttest{#1}{<=}{\arabic{#3depth}}}{%
5823     \LWR@startpars%
```

Create an HTML link to `filename#autofloat-(float number)`, with text of the caption, of the given HTML class.

```
5824     \LWR@subhyperrefclass{%
5825         \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
5826         \#autofloat-\arabic{LWR@nextautofloat}}%
5827         {#4}{toc#2}%
5828     \LWR@stoppars%
5829 }{}%
5830 }
```

Automatically called by `\contentsline`:

```
5831 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
5832 \DeclareDocumentCommand{\l@chapter}{m m}
5833     {\hypertoc{0}{chapter}{#1}{#2}}
5834 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
5835 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
5836 \renewcommand{\l@subsubsection}[2]
5837     {\hypertoc{3}{subsubsection}{#1}{#2}}
5838 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
5839 \renewcommand{\l@ subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
5840 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
5841 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}
```

5842 \end{warpHTML}

62 Index and glossary

See:

[http://tex.stackexchange.com/questions/187038/
how-to-mention-section-number-in-index-created-by-imakeidx](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: 5843 \begin{warpHTML}

5844 \newcounter{LWR@autoindex}
5845 \setcounter{LWR@autoindex}{0}
5846
5847 \newcounter{LWR@autoglossary}
5848 \setcounter{LWR@autoglossary}{0}

\printindex

5849 \let\LWR@origprintindex\printindex
5850
5851 \renewcommand*\printindex{
5852 {
5853 \LWR@startpars
5854 \LWR@origprintindex
5855 }

Env theindex

5856 @ifundefined{chapter}
5857 {\newcommand*{\LWR@indexsection}[1]{\section*{#1}}}
5858 {\newcommand*{\LWR@indexsection}[1]{\chapter*{#1}}}
5859
5860 \renewenvironment*{theindex}%
5861 {\LWR@indexsection{\indexname}}%
5862 {\let\item\LWR@indexitem}%
5863 {\let\subitem\LWR@indexsubitem}%
5864 {\let\subsubitem\LWR@indexsubsubitem}%
5865 }{}
```

\LWR@indexitem

```
5866 \newcommand{\LWR@indexitem}{%
5867
```

```
5868 \InlineClass{indexitem}{}
5869 }

\LWR@indexitem

5870 \newcommand{\LWR@indexsubitem}{%
5871
5872 \InlineClass{indexsubitem}{}
5873 }

\LWR@indexitem

5874 \newcommand{\LWR@indexsubsubitem}{%
5875
5876 \InlineClass{indexsubsubitem}{}
5877 }

\@wrindex {⟨term⟩} Redefined to write the LWR@latestautopage counter instead of page

5878 \def\@wrindex#1{%
5879 \addtocounter{LWR@autoindex}{1}%
5880 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
5881 \protected@write\@indexfile{}}%
5882 {\string\indexentry{#1}{\theLWR@autoindex}}%
5883 \endgroup
5884 \esphack}

\@wrglossary {⟨term⟩} Redefined to write the LWR@latestautopage counter instead of page

5885 \def\@wrglossary#1{%
5886 \addtocounter{LWR@autoglossary}{1}%
5887 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
5888 \protected@write\@glossaryfile{}}%
5889 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
5890 \endgroup
5891 \esphack}

\hyperindexref {⟨autosecnumber⟩}

\hyperindexref{web address} is inserted into *.ind by the xindy style file lwarp.xdy

5892 \newcommand*{\hyperindexref}[1]{\nameref{LWRindex-#1}%

5893 \end{warpHTML}}
```

for PRINT output: A null command for print mode, in case hyperref was not used:

```
5894 \begin{warpprint}
5895 \newcommand{\hyperindexref}[1]{#1}
5896 \end{warpprint}
```

for HTML & PRINT: For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
5897 \begin{warpall}
5898 \providecommand{\glo@name}{}%
5899 \end{warpall}
```

63 Math

- ⚠ Equation numbering** ntheorem has a bug with equation numbering in $\mathcal{AM}\mathcal{S}$ 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.
- Math rendering** Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.
- SVG math option** For SVG math, math is rendered as usual by L^AT_EX into the initial PDF file using the current font¹³, 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 L^AT_EX 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 L^AT_EX code which generated the math, allowing copy/paste of the L^AT_EX math expression into other documents.
- SVG image font size** 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*, for ex:
`\renewcommand{\LateximageFontSizeName}{large}`
- SVG files** As currently implemented, each instance of math creates a new SVG file. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.
- SVG inline** Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images, but dis-allows the possibility of sharing one file among many instances without user intervention.

¹³See section 229 regarding fonts and fractions.

PNG files Others 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 also larger as well.

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 8 regarding EPUB output with MathJax.

MathJax math option The popular MathJax alternative (mathjax.org) may be used to display math.

Prog MathJax

When MathJax is enabled, math is rendered twice:

1. As regular \TeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of \TeX , and
2. As detokenized printed \TeX 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 \TeX values, and the MathJax cross-referencing system is ignored in favor of the \TeX internal system, seamlessly integrating with the rest of the \TeX code.

MathJax limitations Limitations when using MathJax include:

Prog MathJax

chapter numbers

- In document classes which have chapters, \tagged equations have the chapter number prepended in HTML output, unlike \TeX . \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 \TeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

lateximage

- Footnotes inside equations are not yet supported while using MathJax.

- Math appearing inside a lateximage, and therefore also inside a Tikz 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 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/mathjax/MathJax-third-party-extensions/tree/master/siunitx>

⚠ siunitx inside an equation

- \TeX macros**
- MathJax does not automatically support custom \TeX macros, but they may be set up by the user.

custom MathJax macros As an example of using custom \TeX macros with MathJax, place the following at the start of the document, after \begin{document}:

```
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
  % New macros for MathJax are placed inside a math expression:
  \newcommand{\expval}[1]{\langle #1\rangle}
  \newcommand{\abs}[1]{\lvert#1\rvert}
}
\end{warpHTML}
```

for HTML output: 5900 \begin{warpHTML}

\\$ Plain dollar signs appearing in the HTML output may be interpreted by MathJax to be math shifts. For a plain text dollar \\$, print it inside a span to avoid it being interpreted by MathJax, unless are inside a lateximage, in which case it will not be seen by MathJax.

```
5901 \let\LWR@origtextdollar\
5902
5903 \renewcommand*{\$}{%
5904 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
5905 {\LWR@origtextdollar}%
5906 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
5907 }}
```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```
5908 \newcounter{LWR@externalfilecnt}
```

63.1 Inline and display math

```
5909 \LetLtxMacro{\LWR@origdollar$}%
5910 \LetLtxMacro{\LWR@secondorigdollar$}{% balance for editor syntax highlighting
5911 \LetLtxMacro{\LWR@origopenparen}{(}
5912 \LetLtxMacro{\LWR@origcloseparen}{)}
```

```
5913 \LetLtxMacro{\LWR@origopenbracket}{[}
5914 \LetLtxMacro{\LWR@origclosebracket}{]}
```

\$ Redefine the dollar sign to place math inside a `lateximage`, or use MathJax:
 \$\$

```
5915 \begingroup
5916 \catcode`\$=\active%
5917 \protected\gdef\$`{@ifnextchar${\LWR@doubledollar}{\LWR@singledollar}}%
```

`\LWR@doubledollar` Redefine the double dollar sign to place math inside a `lateximage`, or use MathJax:

```
5918 \gdef\LWR@doubledollar##1$${
5919 \ifbool{mathjax}{}
```

For MathJax, print the math between `\[` and `\]`:

```
5920 {\textbackslash[\LWR@HTMLsanitize[#1]\textbackslash]}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the `\textrm` code:

```
5921 {%
  \notmathjax
  \begin{lateximage}
    [\textbackslash[\LWR@HTMLsanitize[#1]\textbackslash]]%
    \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
  \end{lateximage}
}
5929 }%
```

`\LWR@singledollar` Redefine the single dollar sign to place math inside a `lateximage`, or use MathJax:

```
5930 \gdef\LWR@subsingledollar#1{%
  \ifbool{mathjax}{}
```

For MathJax, print the math between `\(` and `\)`:

```
5932 {\textbackslash(\LWR@HTMLsanitize[#1]\textbackslash)}
```

For SVG, print the math inside a `lateximage`, with an ALT tag of the `\textrm` code:

```
5933 {%
  \notmathjax
  \begin{lateximage}
    [\textbackslash(\LWR@HTMLsanitize[#1]\textbackslash)]%
    \LWR@origdollar#1\LWR@origdollar%
  \end{lateximage}
}
5939 }%
```

```

5940
5941 \gdef\LWR@singledollar#1${%
5942 \LWR@subsingledollar{#1}%
5943 }

```

```

\(\ Redefine to the above dollar macros.
\)
5944 \gdef\(#1\){$#1$}
5945 \gdef\[#1\]{$$#1$$}
5946
5947 \endgroup
5948
5949
5950

```

\@ensuredmath {*expression*} Not yet adapted to \warp.

```

5951 \let\LWR@orig@ensuredmath\@ensuredmath
5952
5953 \newcommand*{\LWR@ensuredmath}[1]{%
5954 \LWR@traceinfo{@ensuredmath: page \thepage: \detokenize{#1}}%
5955 \LWR@subsingledollar{#1}%
5956 }

```

Remove the old `math` and `displaymath` environments:

```

5957 \let\math\relax
5958 \let\endmath\relax
5959 \let\displaymath\relax
5960 \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.

```
5961 \NewEnviron{math}{\expandafter\(\BODY\)}
```

`Env displaymath` Set math mode then typeset the body of what was between the begin/end. See the `environ` package for \BODY.

```
5962 \NewEnviron{displaymath}{\expandafter[\BODY]\@ignoretrue}
```

63.2 MathJax support

`Ctr LWR@nextequation` Used to add one to compute the next equation number.

5963 \newcounter{LWR@nextequation}

\LWR@syncmathjax Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “ $\left($ ” and “ $\right)$ ” characters. They are printed to **HTML** output, not interpreted by **LATEX**.

5964 \newcommand*\LWR@syncmathjax{}%

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
5965 \ifcsdef{thechapter}{
5966   \InlineClass{hidden}{
5967     \textbackslash(
5968     \textbackslash) \seteqsection {\thechapter}
5969     \textbackslash)
5970   }
5971 }
5972 {}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
5973 \ifthenelse{\cnttest{\value{equation}}>0}
5974 {
```

Tell MathJax that the next set of equations begins with the current **LATEX** equation number, plus one.

```
5975 \setcounter{LWR@nextequation}{\value{equation}}
5976 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “ $\left($ ” and “ $\right)$ ” characters, to be printed to **HTML**, not interpreted by **LATEX**.

```
5977 \InlineClass{hidden}{
5978   \textbackslash(
5979   \textbackslash) \seteqnumber {\arabic{LWR@nextequation}}
5980   \textbackslash)
5981 }
5982 {}% not eq > 0
5983 }
```

\LWR@restoreorigformatting While producing **SVG** math or a **lateximage**, use regular **LATEX** formatting instead of **HTML** tags.

Various packages add to this macro using **\appto**.

```
5984 \newcommand*{\LWR@restoreorigformatting}{%
5985 \LWR@traceinfo{\LWR@restoreorigformatting}%
5986 \LetLtxMacro{\hspace}{\LWR@orighspace}%
5987 \LetLtxMacro{\rule}{\LWR@origrule}%
5988 \let\,\LWR@origcomma% disable HTML short unbreakable space
5989 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
5990 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
5991 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
5992 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
5993 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
5994 \LetLtxMacro{\textit}{\LWR@origtextit}%
5995 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
5996 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
5997 \LetLtxMacro{\textup}{\LWR@origtextup}%
5998 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
5999 \LetLtxMacro{\emph}{\LWR@origemph}%
6000 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
6001 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
6002 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
6003 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
6004 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
6005 \LetLtxMacro{\upshape}{\LWR@origupshape}%
6006 \LetLtxMacro{\slshape}{\LWR@origslshape}%
6007 \LetLtxMacro{\scshape}{\LWR@origscshape}%
6008 \LetLtxMacro{\itshape}{\LWR@origitshape}%
6009 \LetLtxMacro{\em}{\LWR@origem}%
6010 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
6011 \let\sp\LWR@origsp%
6012 \let\sb\LWR@origsb%
6013 \let\textsuperscript{\LWR@origtextsuperscript}%
6014 \let\textsubscript{\LWR@origtextsubscript}%
6015 \let~\LWR@origtilde%
6016 \let\enskip\LWR@origenskip%
6017 \let\quad\LWR@origquad%
6018 \let\qquad\LWR@origqquad%
6019 \LetLtxMacro{\tabular}{\LWR@origtabular}%
6020 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
6021 \LetLtxMacro{\toprule}{\LWR@origtoprule}%
6022 \LetLtxMacro{\midrule}{\LWR@origmidrule}%
6023 \LetLtxMacro{\cmidrule}{\LWR@origcmidrule}%
6024 \LetLtxMacro{\bottomrule}{\LWR@origbottomrule}%
6025 \LetLtxMacro{\addlinespace}{\LWR@origaddlinespace}%
6026 \LetLtxMacro{\morecmidrules}{\LWR@origmorecmidrules}%
6027 \LetLtxMacro{\specialrule}{\LWR@origspecialrule}%
6028 \let\newline\LWR@orignewline%
6029 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
6030 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
6031 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
6032 \let\reflectbox{\LWR@origreflectbox}%
6033 \let\framebox{\LWR@origframebox}%

```

```
6034 \let\makebox\LWR@origmakebox%
6035 \let\fbox\LWRprint@fbox%
6036 \let\fboxBlock\LWRprint@fbox%
6037 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
6038 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
6039 \LetLtxMacro{\minipage}{\LWR@origminipage}%
6040 \let\endminipage\LWR@endminipage%
6041 \LetLtxMacro{\parbox}{\LWR@origparbox}%
6042 %
6043 \LWR@restoreorigaccents%
6044 %
6045 \LWR@FBcancel%
6046 }
```

\LWR@hidelatexequation {*<environment>*} {*<contents>*}

Creates the \LaTeX version of the equation inside an HTML comment.

```
6047 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
6048 \LWR@stopars
6049 \LWR@htmlopencomment
6050
```

Start the \LaTeX math environment inside the HTML comment:

```
6051 \begingroup
6052 \csuse{LWR@orig#1}
```

While in the math environment, restore various commands to their \LaTeX meanings.

```
6053 \LWR@restoreorigformatting
```

See \LWR@htmleqlabel in section 63.4.1.

Print the contents of the equation:

```
6054 #2
```

End the \LaTeX math environment inside the HTML comment:

```
6055 \csuse{LWR@origend#1}
6056 \endgroup
6057
```

Close the HTML comment and resume HTML paragraph handling:

```
6058 \LWR@htmlclosecomment
6059 \LWR@startpars
6060 }
```

\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**.

```
6061 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed “\(” and “\)” characters.

```
6062 \LWR@origtilde\LWR@orignewline
6063 \textbackslash{}begin{\#1\}
```

Print the contents, sanitizing for **HTML** special characters.

```
6064 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\#2}}
```

Close the MathJax environment:

```
6065 \textbackslash{}end{\#1\}
6066 \LWR@orignewline
6067 }
```

63.3 Equation environment

Remember existing **equation** environment:

```
6068 \let\LWR@origequation\equation
6069 \let\LWR@origendequation\endequation
```

Remove existing **equation** environment:

```
6070 \let\equation\relax
6071 \let\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**.

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 `\labels`. The contents are also printed to the `HTML` output for processing by the MathJax script.

```
6072 \NewEnviron{equation}{%
6073
6074 \ifbool{mathjax}
```

MathJax output:

```
6075 {
```

Print commands to syncronize MathJax's equation number and format to the current `LATEX` chapter/section and equation number:

```
6076     \LWR@syncmathjax
```

Print the `LATEX` math inside an `HTML` comment:

```
6077     \LWR@hidelatexequation{equation}{\BODY}
6078 }
```

SVG output: Create the `lateximage` along with an `HTML` ALT tag having an equation number, the `LATEX` equation environment commands, and the contents of the environment's `\BODY`.

```
6079 {%
  not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
6080     \begin{lateximage}[(\theequation) \textbackslash begin\{equation\}]%
6081     \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}%
6082     \textbackslash end\{equation\}]% alt tag
```

Create the actual `LATEX`-formatted equation inside the `lateximage` using the contents of the environment.

```
6083     \LWR@origequation
6084     \BODY% contents collected by NewEnviron
6085     \LWR@origendequation
6086     \end{lateximage}%
6087 }
6088
```

After the environment, if MathJax, print the math to the `HTML` output for MathJax processing:

```
6089 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

63.4 AMS Math environments

63.4.1 Support macros

Bool `LWR@amsmultiline` True if processing a multiline environment.

To compensate for `multiline`-specific code, `LWR@amsmultiline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if it is used in an `amsmath` environment which is not a `multiline` environment and not an equation.

```
6090 \newbool{LWR@amsmultiline}
6091 \boolfalse{LWR@amsmultiline}
```

`\LWR@htmlmathlabel {<label>}`

`l warp` points `\ltx@label` here. This is used by `\label` when inside a `\TeX` AMS math environment's math display environment.

`\LWR@origltx@label` points to the `\TeX` original, modified by `l warp`, then by `amsmath`, then by `cleveref`.

```
6092 \newcommand*{\LWR@htmlmathlabel}[1]{%
6093 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
6094 \ifbool{mathjax}{%
```

The combined `\TeX` & HTML label is printed in a `\text` field:

```
6095 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6096 \ifbool{LWR@amsmultiline}{}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the `\TeX` & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the `amsmath` package.)

```
6097 \LWR@htmlclosecomment%
6098 \LWR@origltx@label{#1}%
6099 \LWR@htmlopencomment%
6100 }% text
6101 }% mathjax
6102 }%
6103 \LWR@origltx@label{#1}%
6104 }%
6105 }
```

\LWR@beginhideamsmath Starts hiding \LaTeX math inside an HTML comment.

```

6106 \newcommand*{\LWR@beginhideamsmath}{
6107 \LWR@stoppars
6108 \LWR@origtilde\LWR@orignewline
6109 \LWR@htmlopencomment
6110
6111 \begingroup
6112 \LWR@restoreorigformatting
6113 }
```

\LWR@endhideamsmath Ends hiding \LaTeX math inside an HTML comment.

```

6114 \newcommand*{\LWR@endhideamsmath}{
6115 \endgroup
6116
6117 \LWR@htmclosecomment
6118 \LWR@orignewline
6119 \LWR@startpars
6120 }
```

63.4.2 Environment patches

The following amsmath environments already collect their contents in \envbody for further processing. eqnarray is not an $\mathcal{M}\mathcal{S}$ package, and thus requires special handling.

For SVG math: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWR@AMS@BODY` argument telling `lateximage` to use as the HTML ALT tag the environment's contents which were automatically captured by the $\mathcal{M}\mathcal{S}$ 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 environment is not an $\mathcal{M}\mathcal{S}$ environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into \BODY.

```

6121 \let\LWR@origeqnarray\eqnarray
6122 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6123 \newbool{LWR@numbereqnarray}
```

```
6124 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
6125 \newcommand{\LWR@eqnarrayfactor}{%
```

Different behavior depending on MathJax vs SVG math:

```
6126 \ifbool{mathjax}
6127 {
```

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:

```
6128     \LWR@syncmathjax
6129     \boolfalse{LWR@amsmultiline}
6130     \ifbool{LWR@numbereqnarray}
6131     {
```

If numbering the equations, execute a copy inside an `HTML` comment block:

```
6132         \LWR@beginhideamsmath
6133         \LWR@origeqnarray
6134         \BODY
6135         \LWR@origendeqnarray
6136         \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
6137         \LWR@addmathjax{eqnarray}{\BODY}
6138     }%
6139     {%
6140     not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MathJax:

```
6140     \LWR@addmathjax{eqnarray*}{\BODY}
6141     }%
6142     LWR@numbereqnarray
6143 }%
6144     mathjax
6145     {%
6146     not mathjax
6147     \ifbool{LWR@numbereqnarray}
```

For numbered SVG equations, first create a `lateximage` with an `alt` attribute containing sanitized copy of the source code:

```
6148     \begin{lateximage}[\LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual `eqnarray`:

```

6147      \LWR@origeqnarray
6148      \BODY
6149      \LWR@origendeqnarray
6150      \end{lateximage}
6151  }%
6152 {%
6153 not LWR@numbereqnarray

```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```

6153      \begin{lateximage}[\LWR@addmathjax{eqnarray*}\{\BODY\}]
6154      \LWR@origeqnarray
6155      \BODY
6156      \nonumber
6157      \LWR@origendeqnarray
6158      \end{lateximage}
6159  }% LWR@numbereqnarray
6160 }% not mathjax

```

Default to number equations in the future:

```

6161 \booltrue{LWR@numbereqnarray}
6162 }

```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```

6163 \RenewEnviron{eqnarray}
6164 {%
6165
6166 \LWR@eqnarrayfactor
6167
6168 }

```

The starred version is patched to turn off the numbering:

```

6169 \csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

The following *AMS* environments are more easily patched in-place:

```

Env  multiline
      \BeforeBeginEnvironment{multiline}{

6170 \BeforeBeginEnvironment{multiline}{

6171
6172 \ifbool{mathjax}
6173 {

```

```
6174     \LWR@syncmathjax
6175     \booltrue{LWR@amsmultiline}
6176     \LWR@beginhideamsmath
6177 }
6178 {
6179     \lateximage[LWRAMSMATHBODY]
6180 }
6181 }
6182
6183 \AfterEndEnvironment{multiline}{
6184
6185 \ifbool{mathjax}
6186 {
6187     \LWR@endhideamsmath
6188     \boolfalse{LWR@amsmultiline}
6189     \LWR@addmathjax{multiline}{\the\@envbody}
6190 }
6191 {\endlatexitimage}
6192
6193 }

Env  multiline*

6194 \BeforeBeginEnvironment{multiline*}{

6195
6196 \ifbool{mathjax}
6197 {
6198     \LWR@syncmathjax
6199     \booltrue{LWR@amsmultiline}
6200     \LWR@beginhideamsmath
6201 }
6202 {
6203     \latexitimage[LWRAMSMATHBODY]
6204 }
6205 }
6206
6207 \AfterEndEnvironment{multiline*}{

6208
6209 \ifbool{mathjax}
6210 {
6211     \LWR@endhideamsmath
6212     \boolfalse{LWR@amsmultiline}
6213     \LWR@addmathjax{multiline*}{\the\@envbody}
6214 }
6215 {\endlatexitimage}
6216
6217 }
6218
```

Env **gather**

```
6219 \BeforeBeginEnvironment{gather}{

6220 \ifbool{mathjax}

6221 {

6222 \LWR@syncmathjax
6223 \boolearnfalse{\LWR@amsmultiline}
6224 \LWR@beginhideamsmath
6225
6226 }
6227 {
6228 \latexitimage[LWRAMSMATHBODY]
6229 }
6230 }
6231
6232 \AfterEndEnvironment{gather}{

6233 \ifbool{mathjax}

6234 \LWR@endhideamsmath
6235 \LWR@addmathjax{gather}{\the\@envbody}
6236
6237
6238 }
6239 {\endlatexitimage}
6240
6241 }
```

Env **gather***

```
6242 \BeforeBeginEnvironment{gather*}{

6243 \ifbool{mathjax}

6244 \LWR@syncmathjax
6245 {
6246 \LWR@syncmathjax
6247 \boolearnfalse{\LWR@amsmultiline}
6248 \LWR@beginhideamsmath
6249 }
6250 {
6251 \latexitimage[LWRAMSMATHBODY]
6252 }
6253 }
6254
6255 \AfterEndEnvironment{gather*}{

6256 \ifbool{mathjax}

6257 \LWR@endhideamsmath
6258 \LWR@addmathjax{gather*}{\the\@envbody}
6259
6260
6261 }
6262 {\endlatexitimage}
```

```
6263
6264 }

Env  align

6265 \BeforeBeginEnvironment{align}{
6266
6267 \ifbool{mathjax}
6268 {
6269     \LWR@syncmathjax
6270     \boolexpr{LWR@amsmultiline}
6271     \LWR@beginhideamsmath
6272 }
6273 {
6274     \latextimage[LWRAMSMATHBODY]
6275 }
6276 }
6277
6278 \AfterEndEnvironment{align}{

6279 \ifbool{mathjax}
6280 {
6281     \LWR@endhideamsmath
6282     \LWR@addmathjax{align}{\the\@envbody}
6283 }
6284
6285 {\endlatextimage}
6286
6287 }
```

Env align*

```
6288 \BeforeBeginEnvironment{align*}{

6289 \ifbool{mathjax}
6290 {
6291     \LWR@syncmathjax
6292     \boolexpr{LWR@amsmultiline}
6293     \LWR@beginhideamsmath
6294 }
6295
6296 {
6297     \latextimage[LWRAMSMATHBODY]
6298 }
6299 }
6300
6301 \AfterEndEnvironment{align*}{

6302 \ifbool{mathjax}
6303 {
6304 }
```

```
6305      \LWR@endhideamsmath
6306      \LWR@addmathjax{align*}{\the\@envbody}
6307 }
6308 {\endltximage}
6309
6310 }

Env flalign
6311 \BeforeBeginEnvironment{flalign}{
6312
6313 \ifbool{mathjax}
6314 {
6315      \LWR@syncmathjax
6316      \boolearnfalse{\LWR@amsmultiline}
6317      \LWR@beginhideamsmath
6318 }
6319 {
6320      \ltximage[LWRAMSMATHBODY]
6321 }
6322 }
6323
6324 \AfterEndEnvironment{flalign}{

6325
6326 \ifbool{mathjax}
6327 {
6328      \LWR@endhideamsmath
6329      \LWR@addmathjax{flalign}{\the\@envbody}
6330 }
6331 {\endltximage}
6332
6333 }

Env flalign*
6334 \BeforeBeginEnvironment{flalign*}{

6335
6336 \ifbool{mathjax}
6337 {
6338      \LWR@syncmathjax
6339      \boolearnfalse{\LWR@amsmultiline}
6340      \LWR@beginhideamsmath
6341 }
6342 {
6343      \ltximage[LWRAMSMATHBODY]
6344 }
6345
6346
```

```

6347 \AfterEndEnvironment{flalign*}{

6348
6349 \ifbool{mathjax}
6350 {
6351     \LWR@endhideamsmath
6352     \LWR@addmathjax{flalign*}{\the\@envbody}
6353 }
6354 {\endlateximage}
6355
6356 }

6357 \end{warpHTML}

```

64 Lateximages

A `\latexitimage` is typeset on its own PDF page inside an HTML comment which starts on the preceding page and ends on following page, and instructions are written to `latexitimage.txt` for `lwarpmk` to extract the `\latexitimage` 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 `\LWR@` 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.

SVG image font size The size of the math and text used in the `.svg` image may be adjusted by setting `\LatexitimageFontSizeName` to a font size name — *without the backslash*, for ex:
`\renewcommand{\LatexitimageFontSizeName}{large}`

for HTML output: 6358 `\begin{warpHTML}`

Ctr `LWR@latexitimagenumber` Sequence the images.

```

6359 \newcounter{LWR@latexitimagenumber}
6360 \setcounter{LWR@latexitimagenumber}{0}

```

Ctr `LWR@latexitimagedepth` Do not create `\latexitimage` inside of `\latexitimage`.

```

6361 \newcounter{LWR@latexitimagedepth}
6362 \setcounter{LWR@latexitimagedepth}{0}

```

A few utility macros to write special characters:

```
6363 \edef\LWR@hashmark{\string#} % for use in \write
6364 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctrl LWR@LIpage Used to reference the PDF page number of a lateximage to be written into `lateximages.txt`.

```
6365 \newcounter{LWR@LIpage}
```

```
6366 \end{warpHTML}
```

for HTML & PRINT: 6367 \begin{warpall}

\LateximageFontSizeName Declares how large to write text in the `\latexitimage`. The `.svg` file text size should blend well with the surrounding HTML text size.

Do not include the leading backslash in the name.

```
6368 \newcommand*\LateximageFontSizeName[1]{large}
```

```
6369 \end{warpall}
```

for HTML output: 6370 \begin{warpHTML}

\LWR@HTMLsanitize {<text>}

Math expressions are converted to `latexitimage`, 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...

```
6371 \newcommand{\LWR@HTMLsanitize}[1]{%
6372 \begingroup%
6373 \LWR@FBcancel%
6374 \protect\StrSubstitute{\detokenize{#1}}{%
6375 {\detokenize{&}}}{%
6376 {\detokenize{&#38;}}}{[\LWR@strresult]}{%
6377 \protect\StrSubstitute{\detokenize{\expandafter{\LWR@strresult}}}{%
6378 {\detokenize{<}}}{%
6379 {\detokenize{&lt;}}}{%
6380 [\LWR@strresult]}{%
6381 \protect\StrSubstitute{\detokenize{\expandafter{\LWR@strresult}}}{%
6382 {\detokenize{>}}}{%
6383 {\detokenize{&gt;}}}{%
6384 [\LWR@strresult]}{%
6385 \protect\StrSubstitute{\detokenize{\expandafter{\LWR@strresult}}}{%
```

```

6386 {\detokenize{##}}%
6387 {\#}%
6388 [\LWR@strresult]%
6389 \LWR@strresult%
6390 \endgroup%
6391 }

```

\LWR@HTMLsanitizeexpand {*text*}

This version expands the argument before sanitizing it.

```

6392 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
6393 \begingroup%
6394 \LWR@FBcancel%
6395 \protect\StrSubstitute{\detokenize\expandafter{\#1}}{%
6396 {\detokenize{&}}{%
6397 {\detokenize{&#}}{%
6398 [\LWR@strresult]%
6399 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
6400 {\detokenize{<}}{%
6401 {\detokenize{<}}{%
6402 [\LWR@strresult]%
6403 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
6404 {\detokenize{>}}{%
6405 {\detokenize{>}}{%
6406 [\LWR@strresult]%
6407 \LWR@strresult%
6408 \endgroup%
6409 }

```

Env lateximage [*altnode*]

```

6410 \catcode`\$=\active%
6411
6412 \NewDocumentEnvironment{lateximage}{O{image}}
6413 {%
6414 \LWR@traceinfo{lateximage: starting on page \arabic{page}}{%
6415 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

6416 {%
6417     \addtocounter{\LWR@lateximagedepth}{1}%
6418 }%

```

Otherwise, this is the outer-most lateximage:

```

6419 {% start of outer-most lateximage

```

Starting a new lateximage:

```
6420 \addtocounter{LWR@lateximagenumber}{1}%
6421 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
6422 \boolfalse{mathjax}
```

Be sure that are doing a paragraph:

```
6423 \LWR@ensuredoingapar%
```

Next file:

```
6424 \addtocounter{LWR@externalfilecnt}{1}%
6425 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
6426 \setcounterpageref{LWR@LIpage}{LWR@lateximage\theLWR@lateximagenumber}%
6427 \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:

```
6428 \LWR@htmlltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
6429 class="lateximagesource"{} } \LWR@orignewline
```

Write instructions to the lateximages.txt file:

```
6430 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
6431 \immediate\write\LWR@lateximagesfile{| \theLWR@LIpage | \theLWR@externalfilecnt |}%
```

Place an open comment tag at the bottom of page; footnotes will be above this tag. This will hide any traces of the lateximage PDF page which were picked up by pdftotext.

```
6432 \LWR@traceinfo{lateximage: about to create open comment}%
6433 \LWR@htmlopencomment%
```

One level deeper:

```
6434 \addtocounter{LWR@latimedepth}{1}%
```

Start the new PDF page:

```
6435     \LWR@traceinfo{lateximage: about to create new page}%
6436     \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
6437     \LWR@traceinfo{lateximage: about to create minipage}%
6438     \LWR@origminipage{6in}%
6439     \csuse{\LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \\hspace, etc. inside a lateximage.

```
6440     \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
6441     \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
6442     \def\@mpfn{footnote}%
6443     \def\thempfn{\thefootnote}%
6444     \let\@footnotetext\LWR@footnotetext%
```

Create the LWRlateximage<number> label:

```
6445     \LWR@traceinfo{lateximage: about to create label}%
6446     \LWR@origlabel{\LWRlateximage\arabic{\LWR@lateximagenumber}}%
6447     \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
6448     \LetLtxMacro$\LWR@origdollar%
6449     \catcode`$=3% math shift
6450     \LetLtxMacro(\LWR@origopenparen%
6451     \LetLtxMacro)\LWR@origcloseparen%
6452     \LetLtxMacro[\LWR@origopenbracket%
6453     \LetLtxMacro]\LWR@origclosebracket%
6454     \let\@ensuredmath\LWR@orig@ensuredmath%
6455 }% end of outer-most lateximage
6456 \LWR@traceinfo{lateximage: finished start of environment}%
6457 }% end of \begin{lateximage}
```

When the environment closes:

```
6458 {%
6459 \LWR@traceinfo{lateximage: starting end of environment}%
}
```

Nested more than one deep?

```
6460 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6461 {\addtocounter{LWR@lateximagedepth}{-1}}%
```

If this is the outer-most lateximage:

```
6462 { % end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
6463 \LWR@origendminipage%
6464 \LWR@orignewline%
6465 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
6466 \LWR@htmclosecomment{} \LWR@orignewline%
6467 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the amsmath multiline, gather, or align environments.

```
6468 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%
6469 { %
6470 \LWR@subinlineimage[%
6471 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
6472 ]%
6473 {lateximage}%
6474 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}%
6475 {svg}%
6476 {}%
6477 ]%
6478 { %
6479 \LWR@subinlineimage[#1]{lateximage}%
6480 {lateximages\OSPathSymbol{}lateximage-\theLWR@externalfilecnt}{svg}{}%
6481 ]%
6482 % \LWR@orignewline% Removed to prevent extra space.
```

Be sure that are doing a paragraph:

```
6483 \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```
6484     \LWR@htmltag{/span}%
6485     \ifbool{HTMLDebugComments}{%
6486         \LWR@htmlcomment{End of lateximage}%
6487     }{}%
6488 \% \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
6489     \addtocounter{LWR@lateximagedepth}{-1}%
6490 }% end of outer-most lateximage
6491 \LWR@traceinfo{lateximage: done}%
6492 }%
6493 \catcode`\$=3% math shift
6494 \end{warpHTML}
```

for PRINT output: 6495 \begin{warpprint}
6496 \% \newenvironment{lateximage}[1][]{\minipage{\linewidth}}{\endminipage}
6497 \newenvironment{lateximage}[1][]{\begin{array}{c}}
6498 \end{warpprint}

65 center, flushleft, flushright

for HTML output: 6499 \begin{warpHTML}

Env center Replace center functionality with css tags:

```
6500 \renewenvironment*{center}%
6501 {%
6502 \LWR@forcenewpage
6503 \BlockClass{center}%
6504 }%
6505 {\endBlockClass}
```

Env flushright

```
6506 \renewenvironment*{flushright}%
6507 {%
6508 \LWR@forcenewpage
6509 \BlockClass{flushright}%
6510 }%
6511 {\endBlockClass}
```

```
Env   flushleft

6512 \renewenvironment*{flushleft}
6513 {
6514 \LWR@forcenewpage
6515 \BlockClass{flushleft}
6516 }
6517 {\endBlockClass}

6518 \end{warpHTML}
```

66 Textcomp

for HTML output: 6519 \begin{warpHTML}

If textcomp was loaded before l warp, perhaps as part of the font-related packages, explicitly load the l warp patches now:

```
6520 \@ifpackageloaded{textcomp}
6521 {
6522 \LWR@origRequirePackage{l warp-textcomp}
6523 }
6524 {}

6525 \end{warpHTML}
```

67 Siunitx

Pkg siunitx

⚠ per-mode Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

Some units will require that the expression be placed inside math mode.

for HTML output: 6526 \begin{warpHTML}

Options for siunitx:

```
6527 \PassOptionsToPackage{
6528   detect-mode=true,
6529   per-mode=symbol,% fraction is not seen by pdftotext
6530 %   text-celsius = {\protect\LWRsiunitx@degree{}C},
```

```

6531 %     text-degree = {\protect\LWRsiunitx@degree},
6532 }{\siunitx}
6533
6534 \end{warpHTML}
```

68 Graphics

Pkg `graphics`
Pkg `graphicx`

- ⚠ **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 are not supported by `warp`; the entire image will be shown.
- ⚠ **\graphicspath** `\graphicspath` only works for a single directory; all graphics must be in this directory.
- 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. Using 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.
- 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.
- ⚠ **image file types** For `\includegraphics` the user should provide both `.pdf` and `.svg` images, but always refer to `.pdf` images in the document source. All `\includegraphics` references to `.pdf` will automatically be changed to `.svg` for `HTML` output, and will be left as `.pdf` for print output. Images may also be `.jpg` and `.png`, and will be used as-is for either output.
- \rotatebox** `\rotatebox` accepts the optional `origin` key.
- ⚠ **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The css standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike L^AT_EX, so expect some ugly results for scaling and rotating.
- for HTML output:** 6535 `\begin{warpHTML}`

68.1 \graphicspath

```
\graphicspath {<path>}
6536 \newcommand*{\thisgraphicspath}{{}}
6537 \renewcommand*{\graphicspath}[1]{\renewcommand*{\thisgraphicspath}{#1}}
\DeclareGraphicsExtensions {<list>}
\DeclareGraphicsRule {}{}{}{}{<list>}
6538 \renewcommand*{\DeclareGraphicsExtensions}[1]{}
6539 \renewcommand*{\DeclareGraphicsRule}[4]{}
```

68.2 Length conversions and graphics options

⚠ whitespace A scaled image in L^AT_EX 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.

```
6540 \renewcommand*{\unitspace}{}
```

Used to store the user's selected dimensions and HTML class.

The class defaults to "inlineimage" unless changed by a `class=xyx` option.

```
6541 \newlength{\LWR@igwidth}
6542 \newlength{\LWR@igheight}
6543 \newcommand*{\LWR@igwidthstyle}{{}}
6544 \newcommand*{\LWR@igheightstyle}{{}}
6545 \newcommand*{\LWR@igorigin}{{}}
6546 \newcommand*{\LWR@igangle}{{}}
6547 \newcommand*{\LWR@igxscale}{1}
6548 \newcommand*{\LWR@igyscale}{1}
6549 \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:

```
6550 \define@key{igraph}{width}{%
6551 \setlength{\LWR@igwidth}{#1}%
6552 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
6553 {%
```

Default to use the converted fixed length given:

```
6554 \uselengthunit{PT}%
6555 \renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
6556 \IfEndWith{#1}{ex}%
6557 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes ex
6558 {}% not ex
6559 \IfEndWith{#1}{em}%
6560 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes em
6561 {}% not em
6562 \IfEndWith{#1}{\%}%
6563 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes percent
6564 {}% not percent
6565 \IfEndWith{#1}{px}%
6566 {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
6567 {}% not px
6568 }{}% end of length > Opt
6569 }
```

If an optional height was given, set an HTML style:

```
6570 \define@key{igraph}{height}{%
6571 \setlength{\LWR@igheight}{#1}%
6572 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}%
6573 {}%
```

Default to use the converted fixed length given:

```
6574 \uselengthunit{PT}%
6575 \renewcommand*{\LWR@igheightstyle}{%
6576 height:\rndprintlength{\LWR@igheight} %
6577 }%
```

If ex or em dimensions were given, use those instead:

```
6578 \IfEndWith{#1}{ex}%
6579 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
6580 {}% not ex
6581 \IfEndWith{#1}{em}%
6582 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
6583 {}% not em
6584 \IfEndWith{#1}{\%}%
6585 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
6586 {}% not percent
6587 \IfEndWith{#1}{px}%
6588 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
```

```

6589      {}% not px
6590 }{}% end of length > Opt
6591 }

```

Handle origin key:

```

6592 \define@key{igraph}{origin}{%
6593 \renewcommand*{\LWR@igorigin}{#1}%
6594 }

```

Handle angle key:

```
6595 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}
```

Handle class key:

```

6596 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
6597

```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```

6598 \define@key{igraph}{scale}{%
6599 \renewcommand*{\LWR@igxscale}{#1}%
6600 \renewcommand*{\LWR@igyscale}{#1}}

```

Numerous ignored keys:

```

6601 \define@key{igraph}{bb}{}
6602 \define@key{igraph}{bbllx}{}
6603 \define@key{igraph}{bbly}{}
6604 \define@key{igraph}{bburx}{}
6605 \define@key{igraph}{bbury}{}
6606 \define@key{igraph}{natwidth}{}
6607 \define@key{igraph}{natheight}{}
6608 \define@key{igraph}{hiresbb}{}
6609 \define@key{igraph}{viewport}{}
6610 \define@key{igraph}{trim}{}
6611 \define@key{igraph}{totalheight}{}
6612 \define@key{igraph}{keepaspectratio}{}
6613 \define@key{igraph}{clip}{}
6614 \define@key{igraph}{draft}{}
6615 \define@key{igraph}{type}{}
6616 \define@key{igraph}{ext}{}
6617 \define@key{igraph}{read}{}
6618 \define@key{igraph}{command}{}

```

```
\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.

```
6619 \newcommand*{\LWR@rotstyle}[2]{%
6620 #1transform:rotate(-#2deg);%
6621 }
```

`\LWR@scaletyle {<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.

```
6622 \newcommand*{\LWR@scaletyle}[3]{%
6623 #1transform:scale(#2,#3);%
6624 }
```

68.3 \includegraphics

`Bool LWR@infloatrow` Used to compute `\linewidth`.

```
6625 \newbool{\LWR@infloatrow}
6626 \boolfalse{\LWR@infloatrow}

\LWR@opacity may be set by the transparent package. For HTML it is only used for
\includegraphics.

6627 \def\LWR@opacity{1}
```

```
6628 \newcommand*{\LWR@imageextension}{}%
6629 \newcommand*{\LWR@expgraphicsfilename}{}%
```

`\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.

```
6630 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}%
6631 {%
```

Start the image tag on a new line, allow PDF output word wrap:

```
6632 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
6633 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
6634 {%
6635   \ifbool{LWR@infloatrow}%
6636   {}%
6637   {%
6638     \setlength{\linewidth}{6in}%
6639     \setlength{\textwidth}{6in}%
6640     \setlength{\textheight}{9in}%
6641   }%
6642 }{}%
```

See if can find the image by adding an extension:

Preference is `svgz`, then `svg`, `gif`, `png`, and `jpg`.

`\detokenize\expandafter` allows underscore characters in filenames.

```
6643 \edef\LWR@expgraphicsfilename{#4}
6644 \renewcommand*\LWR@imageextension{}%
6645 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.jpg}%
6646 {\renewcommand*\LWR@imageextension{.jpg}{}%}
6647 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.JPG}%
6648 {\renewcommand*\LWR@imageextension{.JPG}{}%}
6649 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.png}%
6650 {\renewcommand*\LWR@imageextension{.png}{}%}
6651 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.PNG}%
6652 {\renewcommand*\LWR@imageextension{.PNG}{}%}
6653 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.gif}%
6654 {\renewcommand*\LWR@imageextension{.gif}{}%}
6655 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.GIF}%
6656 {\renewcommand*\LWR@imageextension{.GIF}{}%}
6657 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.svg}%
6658 {\renewcommand*\LWR@imageextension{.svg}{}%}
6659 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.SVG}%
6660 {\renewcommand*\LWR@imageextension{.SVG}{}%}
6661 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.svgz}%
6662 {\renewcommand*\LWR@imageextension{.svgz}{}%}
6663 \IfFileExists{\detokenize\expandafter>thisgraphicspath\LWR@expgraphicsfilename.SVGZ}%
6664 {\renewcommand*\LWR@imageextension{.SVGZ}{}%}
```

Convert a PDF extension to SVG, leave the result in `\LWR@strresult`:

Must also \detokenize .pdf and .svg comparison strings.

```
6665 \StrSubstitute{\detokenize\expandafter{\LWR@expgraphicsfilename}}%
6666 {\detokenize{.pdf}}{\detokenize{.svg}}[\LWR@strresult]%
6667 %
6668 \StrSubstitute{\LWR@strresult}%
6669 {\detokenize{.PDF}}{\detokenize{.SVG}}[\LWR@strresult]%
```

For correct em sizing during the width and height conversions:

```
6670 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
6671 \setlength{\LWR@igwidth}{Opt}%
6672 \setlength{\LWR@igheight}{Opt}%
6673 \renewcommand*{\LWR@igwidthstyle}{}%
6674 \renewcommand*{\LWR@igheightstyle}{}%
6675 \renewcommand*{\LWR@igorigin}{}%
6676 \renewcommand*{\LWR@igangle}{}%
6677 \renewcommand*{\LWR@igxscale}{1}%
6678 \renewcommand*{\LWR@igyscale}{1}%
6679 \renewcommand*{\LWR@igclass}{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```
6680 \IfValueTF{#3}{}{%
6681 \IfValueTF{#2}{%
6682 {\setkeys{igraph}{#2}}%
6683 {\setkeys{igraph}{}}
6684 }%
```

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.

```
6685 \href{\thisgraphicspath\LWR@strresult\LWR@imageextension}%
6686 {%
6687 \LWR@htmntag{%
6688 img src="\thisgraphicspath\LWR@strresult\LWR@imageextension" \LWR@orignewline
6689 \LWR@origtilde{} alt="\LWR@strresult" \LWR@orignewline}
```

Only include a style tag if a width, height, angle, or scale was given:

```
6690 \ifthenelse{%
6691   \NOT\equal{\LWR@igwidthstyle}{} \OR
6692   \NOT\equal{\LWR@igheightstyle}{} \OR
```

```

6693      \NOT\equal{\LWR@igorigin}{} \OR
6694      \NOT\equal{\LWR@igangle}{} \OR
6695      \NOT\equal{\LWR@igxscale}{1} \OR
6696      \NOT\equal{\LWR@igyscale}{1}
6697 }%
6698 {\LWR@origtilde{} style="%"
6699 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{} }{%
6700 {\LWR@igwidthstyle;}}{%
6701 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{} }{%
6702 {\LWR@igheightstyle;}}{%
6703 \ifthenelse{\NOT\equal{\LWR@igorigin}{} }{%
6704 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline;} }{%
6705 \ifthenelse{\NOT\equal{\LWR@igangle}{} }{%
6706 {%
6707 \LWR@rotstyle{-ms-}{\LWR@igangle}%
6708 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
6709 \LWR@rotstyle{}{\LWR@igangle}%
6710 }}{%
6711 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%}
6712 \NOT\equal{\LWR@igyscale}{1}}{%
6713 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}}%
6714 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
6715 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}}{%
6716 %
6717 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}{%
6718 {\opacity:\LWR@opacity;}}{%
6719 }}{%
6720 %
6721 " \LWR@orignewline;} }%

```

Set the class:

```

6722 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
6723 }% end of image tags
6724 }% end of href
6725 \endgroup

```

Return to small-sized output:

```

6726 \LWR@origscriptsize
6727 }

```

\includegraphics [⟨key=val⟩] {⟨filename⟩}

Handles width and height, converted to fixed width and heights.

Converts any .pdf references to .svg for HTML

The user should always refer to .pdf in the document source.

```
6728 \renewcommand*{\includegraphics}{%
6729 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
6730 \LWR@ensuredoingapar%
6731 \begingroup%
6732 \LWR@includegraphicsb%
6733 }%
6734 \end{warpHTML}
```

for PRINT output: For print output, accept and then discard the new class key:

```
6735 \begin{warpprint}%
6736 \define@key{Gin}{class}{}%
6737 \end{warpprint}
```

68.4 \rotatebox, \scalebox, \reflectbox

for HTML output: 6738 \begin{warpHTML}

\LWR@rotboxorigin Holds the origin key letters.

```
6739 \newcommand*{\LWR@rotboxorigin}{}%
```

\LWR@originname {\langle letter\rangle}

Given one \TeX origin key value, translate into an HTML origin word:

```
6740 \newcommand*{\LWR@originname}[1]{%
6741 \ifthenelse{\equal{#1}{t}}{top}{}%
6742 \ifthenelse{\equal{#1}{b}}{bottom}{}%
6743 \ifthenelse{\equal{#1}{c}}{center}{}%
6744 \ifthenelse{\equal{#1}{l}}{left}{}%
6745 \ifthenelse{\equal{#1}{r}}{right}{}%
6746 }
```

\LWR@originnames {\langle letters\rangle}

Given one- or two-letter \TeX origin key values, translate into HTML origin words:

```

6747 \newcommand*\LWR@originnames[1]{%
6748 \StrChar{#1}{1}[\LWR@strresult]%
6749 \LWR@originname{\LWR@strresult}%
6750 \StrChar{#1}{2}[\LWR@strresult]%
6751 \LWR@originname{\LWR@strresult}%
6752 }

```

Handle the origin key for `\rotatebox`:

```

6753 \define@key{krotbox}{origin}{%
6754 \renewcommand*\LWR@rotboxorigin{#1}%
6755 }

```

These keys are ignored:

```

6756 \define@key{krotbox}{x}{}
6757 \define@key{krotbox}{y}{}
6758 \define@key{krotbox}{units}{}

```

`\rotatebox [⟨keyval list⟩] {⟨angle⟩} {⟨text⟩}`

Will `\let\rotatebox\LWR@rotatebox` at `\LWR@LwarpStart`, in case `\rotatebox` was over-written by a later package load.

```

6759 \LetLtxMacro\LWR@origrotatebox\rotatebox
6760
6761 \NewDocumentCommand{\LWR@rotatebox}{O{} m +m}{%

```

Reset the origin to “none-given”:

```
6762 \renewcommand*\LWR@rotboxorigin{}
```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
6763 \setkeys{krotbox}{#1}%
```

Select `inline-block` so that HTML will transform this span:

```
6764 \LWR@htmllagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```

6765 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}{%
6766 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```

6767 \LWR@rotstyle{-ms-}{#2} %
6768 \LWR@rotstyle{-webkit-}{#2} %
6769 \LWR@rotstyle{}{#2} %
6770 "{}"\LWR@orignewline%

```

Print the text to be rotated:

```

6771 \begin{LWR@nestspan}%
6772 #3%

```

Close the span:

```

6773 \LWR@htmltagc{/span}%
6774 \end{LWR@nestspan}%
6775 }

```

\scalebox {⟨h-scale⟩} [⟨v-scale⟩] {⟨text⟩}

Will \let\scalebox\LWR@scalebox at \LWR@LwarpStart, in case \scalebox was over-written by a later package load.

```

6776 \LetLtxMacro{\LWR@origscalebox}{\scalebox}
6777
6778 \NewDocumentCommand{\LWR@scalebox}{m o m}%

```

Select inline-block so that HTML will transform this span:

```

6779 \LWR@htmltagc{span style="display: inline-block; "

```

Print the scaling information:

```

6780 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6781 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6782 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
6783 "{}"%

```

Print the text to be scaled:

```

6784 \begin{LWR@nestspan}%
6785 #3%

```

Close the span:

```

6786 \LWR@htmltagc{/span}%
6787 \end{LWR@nestspan}%
6788 }

```

\reflectbox {⟨text⟩}

Will \let\reflectbox\LWR@reflectbox at \LWR@LwarpStart, in case \reflectbox was over-written by a later package load.

```
6789 \let\LWR@origreflectbox\reflectbox
6790
6791 \newcommand{\LWR@reflectbox}[1]{\LWR@scalebox{-1}[1]{#1}}
6792 \end{warpHTML}
```

68.5 Null functions

These functions are not supported by lwarp's HTML conversion.

for HTML output: 6793 \begin{warpHTML}

```
\resizebox {<h-length>} {<v-length>} {<text>}
```

Simply prints its text argument.

```
6794 \renewcommand{\resizebox}[3]{#3}
6795 \end{warpHTML}
```

68.6 xcolor print-mode additions

Print-mode versions of new xcolor definitions. 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: 6796 \begin{warpall}

After `xparse` may have been loaded ...

```
6797 \AtBeginDocument{
    ... and only if xcolor was loaded:
6798 \@ifpackageloaded{xcolor}{
6799 \LWR@traceinfo{patching xcolor}}
```

\colorboxBlock \colorboxBlock is the same as \colorbox:

```
6800 \LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```
6801 \warpprintonly{  
6802 \LetLtxMacro\LWRprint@colorboxBlock\colorbox  
6803 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox  
6804 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox  
6805 }
```

```
\fcolorbox [⟨framemode⟩] {⟨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.)

```
6806 \newsavebox{\LWR@colorminipagebox}  
6807  
6808 \DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%  
6809 \LWR@traceinfo{\LWRprint@fcolorbox #2 #4} %
```

Pre-load the contents into an LR box so that they can be used inside a \fcolorbox:

```
6810 \begin{lrbox}{\LWR@colorminipagebox} %  
6811 #5%  
6812 \end{lrbox} %
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a \fcolorbox.

```
6813 \ifthenelse{\equal{#4}{none}}{  
6814 { % #4 none  
6815     \LWR@traceinfo{background is none} %  
6816     { % scope the \colorlet  
6817         \colorlet{\LWR@currentcolor}{.} %  
6818         \color{#2} %  
6819         \fbox{ %  
6820             \color{\LWR@currentcolor} %  
6821             \usebox{\LWR@colorminipagebox} %  
6822         } % fbox  
6823     } % colorlet  
6824 } % #4 none  
6825 { % #4 not none  
6826 \LWR@traceinfo{background not none} %
```

```

6827 \IfValueTF{#1}%
6828 {%
6829   \IfValueTF{#3}%
6830   {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}{%
6831   {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}}%
6832 }%
6833 {%
6834   \IfValueTF{#3}%
6835   {\LWRorigprint@fcolorbox[#2]{#3}{#4}{\usebox{\LWR@colorminipagebox}}}{%
6836   {\LWRorigprint@fcolorbox[#2]{#4}{\usebox{\LWR@colorminipagebox}}}}%
6837 }%
6838 }%
6839 \LWR@traceinfo{LWRprint@fcolorbox done}%
6840 }

```

\fcolorboxBlock [*framemode*] {[*framecolor*] [*boxmodel*] {[*boxcolor*] {[*text*]}}

In print mode, \fcolorboxBlock is the same as \fcolorbox.

```
6841 \LetLtxMacro{\LWRprint@fcolorboxBlock}{\LWRprint@fcolorbox}
```

Env fcolorminipage [*1:framemode*] {[*2:framecolor*] [*3:boxmodel*] {[*4:boxcolor*] [*5:align*] [*6:height*] [*7:inner-align*] {[*8:width*]}}

In print mode, becomes a \fcolorbox containing a minipage:

```

6842 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{} o m}%
6843 {%
6844 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a \fcolorbox:

```
6845 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```

6846 \IfValueTF{#7}%
6847 {\begin{minipage}[#5][#6][#7]{#8}}%
6848 {\begin{minipage}[#5][#6][#5]{#8}}%
6849 }%
6850 {%
6851 \end{minipage}%
6852 \end{lrbox}%
6853 \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.

```

6854 \ifthenelse{\equal{#4}{none}}%
6855 {%
6856   #4 none
6857   {%
6858     \colorlet{LWR@currentcolor}{.}%
6859     \color{#2}%
6860     \fbox{%
6861       \color{LWR@currentcolor}%
6862       \usebox{\LWR@colorminipagebox}%
6863     }%
6864   }%
6865 }%
6866 #4 none
6867 {%
6868 \IfValueTF{#1}%
6869 {%
6870   {\LWRorigprint@fcolorbox[#1][#2][#3][#4]{\usebox{\LWR@colorminipagebox}}}%
6871   {\LWRorigprint@fcolorbox[#1][#2][#4]{\usebox{\LWR@colorminipagebox}}}%
6872 }%
6873 {%
6874   {\LWRorigprint@fcolorbox[#2][#3][#4]{\usebox{\LWR@colorminipagebox}}}%
6875   {\LWRorigprint@fcolorbox[#2][#4]{\usebox{\LWR@colorminipagebox}}}%
6876 }%
6877 }%
6878 }%
6879 }%
6880 \LWR@traceinfo{*** finished end fcolorminipage}%

```

\LWR@restoreorigprintxcolor Used to activate print-mode additions for xcolor. In print mode, this is used immediately following. In HTML mode, this is used inside a `latexitimage`.

```

6880 \newcommand*{\LWR@restoreorigprintxcolor}{%
6881 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
6882 \LetLtxMacro{\colorboxBlock}{\LWRprint@colorboxBlock}%
6883 \LetLtxMacro{\fcolorbox}{\LWRprint@fcolorbox}%
6884 \LetLtxMacro{\fcolorboxBlock}{\LWRprint@fcolorboxBlock}%
6885 \LetLtxMacro{\fcolorminipage}{\LWRprint@fcolorminipage}%
6886 \LetLtxMacro{\endfcolorminipage}{\endLWRprint@fcolorminipage}%
6887 }%
6888 %
6889 \appto{\LWR@restoreorigformatting}{%
6890 \LWR@restoreorigprintxcolor}%
6891 }%

```

If print mode, immediately activate the print-mode enhancements for xcolor:

```

6892 \warpprintonly{\LWR@restoreorigprintxcolor}%
6893 %
6894 \LWR@traceinfo{xcolor patches done}%

```

```
6895 }{}% xcolor loaded
6896 }% AtBeginDocument

6897 \end{warpall}
```

69 Cleveref

Pkg cleveref cleveref package is used as-is with minor patches.

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 l warp 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 l warp's additional loading of cleveref will have no effect.

Table 8 on 295 shows the data structure of the label/reference system as revised by l warp and cleveref.

A few patches allow cleveref to work as-is:

for HTML output: 6898 `\begin{warpHTML}`

`\AtEndPreamble` forces cleveref to be loaded last:

```
6899 \AtEndPreamble{
6900 \RequirePackage{cleveref}
6901 }
```

The following patches are applied after cleveref has loaded, and after `\AtBeginDocument`:

```
6902 \AfterEndPreamble{
6903 \LWR@traceinfo{Patching cleveref.}
```

```
\@@setcref {\langle kindofref\rangle} {\langle label\rangle}
```

```
6904 \renewcommand*\@@setcref[2]{#1{\ref{#2}}{}{}}
```

```
\@@setcrefrange {\langle text\rangle} {\langle label\rangle} {\langle label\rangle}
```

```
6905 \renewcommand{\@@setcrefrange}[3]{%
6906 #1{\ref{#2}}{\ref{#3}}{}{}{}}
```

\cpagerefFor Redefinable word between “page(s)” and the page numbers.

6907 \newcommand*{\cpagerefFor}{for}

\@setcpageref {*typeofref*} {*label*}, where *typeofref* is “page” or “pages”

6908 \renewcommand*{\@setcpageref}[2]{%
6909 #1{\cpagerefFor\ \cref{#2}}{}{}%
6910 }

6911 \renewcommand{\@setcpagerefrange}[3]{%
6912 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{}{}{}{}%
6913 }% AfterEndPreamble

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

6914 \let\LWR@origlabel\label
6915 \let\label\LWR@newlabel
6916 \let\LWR@origref\ref
6917 \let\ref\LWR@newref% \end{ syntax highlighting
6918 \let\LWR@origpageref\pageref
6919 \let\pageref\LWR@newpageref
6920
6921
6922
6923 \end{warpHTML}

70 Picture

Env picture The picture environment is enclosed inside a \lateximage.

for HTML output: 6924 \begin{warpHTML}

Env picture

6925 \BeforeBeginEnvironment{picture}{\lateximage}
6926
6927 \AfterEndEnvironment{picture}{\endlateximage}

6928 \end{warpHTML}

71 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 suppressed 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 `. An `HTML <div>` cannot appear inside a ``. While in a ``, 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 `TEX` PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

for HTML output: 6929 `\begin{warpHTML}`

71.1 Counters and lengths

Ctr `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
6930 \newcounter{LWR@minipagedepth}
6931 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
6932 \newlength{\LWR@minipagewidth}
```

\len \W\minipageheight Used to convert the height into printable units.

```
6933 \newlength{\LWR@minipageheight}
```

71.2 Footnote handling

Also see section 45 for other forms of footnotes. Minipage footnotes are gathered in section 45.5, and then placed into the document in section 71.3.

71.3 Minipage handling

\LWR@endminipage Used to close a minipage.

Copied the L^AT_EX definition and modified to create a <div> of class mpfootnotes:

```
6934 \def\LWR@endminipage{%
6935   \par
6936   \unskip
6937   \ifvoid\@mpfootins\else
6938     \vskip\skip\@mpfootins
6939     \normalcolor
6940     \LWR@htmldivclass{mpfootnotes}
6941     \LWR@origmedskip
6942     \unvbox\@mpfootins
6943     \LWR@htmldivclassend{mpfootnotes}
6944   \fi
6945   \minipagefalse
6946   \color@endgroup
6947   \egroup
6948   \expandafter\iiiparbox\mpargs{\unvbox\@tempboxa}}
```

\LWR@subminipage Used to create a PDF minipage without creating an HTML minipage. This allows footnotes to appear at the bottom of the minipage instead of the bottom of the HTML page.

```
6949 \newcommand*\LWR@subminipage{%
6950 \LWR@stopars
6951 \LWR@origminipage{6in}}
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```
6952 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6953 \LWR@startpars%
6954 }
```

\LWR@endsubminipage Closes the subminipage.

```
6955 \newcommand*\LWR@endsubminipage{%
6956 \LWR@stoppars%
6957 \LWR@endminipage% The following empty line is required:
6958
6959 }
```

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```
6960 \newbool{LWR@minipagefullwidth}
6961 \boolfalse{LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

for HTML output: 6962 \newcommand*\minipagefullwidth{\booltrue{LWR@minipagefullwidth}}
6963 \end{warpHTML}

for PRINT output: 6964 \begin{warpprint}
6965 \newcommand*\minipagefullwidth{}
6966 \end{warpprint}

for HTML output: 6967 \begin{warpHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
6968 \newbool{LWR@minipagethispar}
6969 \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.

```
6970 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
6971 {%
6972 \uselengthunit{in}}%
```

```

6973 \setlength{\LWR@minipagewidth}{#4}%
6974 \ifthenelse{\cnttest{\value{\LWR@minipagedepth}}{=}{0}}{%
6975   \addtolength{\LWR@minipagewidth}{3em}%
6976   \setlength{\linewidth}{6in}%
6977   \setlength{\textwidth}{6in}%
6978   \setlength{\textheight}{9in}%
6979 }{}%
6980 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
6981 \addtocounter{\LWR@minipagedepth}{1}%
6982 \setlength{\LWR@minipageheight}{\textheight}%
6983 \IfValueTF{#2}{\setlength{\LWR@minipageheight}{#2}}{}%

```

\LaTeX wants to start a paragraph for the new minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
6984 \LWR@stopars%
```

Create the <div> tag with optional alignment style:

```

6985 \LWR@traceinfo{minipage: creating div class}%
6986 \LWR@orignewpage%
6987 \LWR@htmlltag{div class="minipage" style="%"
6988 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
6989 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
6990 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
6991 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
6992 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
6993 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
6994 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%

```

Print the width and optional height styles:

```

6995 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
6996 \uselengthunit{PT}%
6997 \ifbool{\LWR@minipagefullwidth}{}%
6998 {\boolefalse{\LWR@minipagefullwidth}}%
6999 {}%
7000 \ifthenelse{\lengthtest{#4}=\linewidth}{%
7001   {}%
7002   {width:\rndprintlength{\LWR@minipagewidth} ; }%
7003 }%
7004 \LWR@traceinfo{minipage: about to print the height}%
7005 \IfValueTF{#2}{height:\rndprintlength{\LWR@minipageheight} ; }{}%
7006 "{}"%

```

Finish with an empty line to start \LaTeX minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```
7007
7008 \LWR@origminipage{6in}%
  The preceding empty line is required.
```

Set the user-accessible minipage and text width and height values inside the minipage. These do not affect the actual size of the large minipage created by \LWR@origminipage above, but are used by any reference to \linewidth, etc. inside the PDF minipage being created here.

```
7009 \setlength{\linewidth}{#4}%
  the original width
7010 \setlength{\textwidth}{6in}%
7011 \setlength{\textheight}{9in}%
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```
7012 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
7013 \LWR@startpars%
7014 \LWR@traceinfo{minipage: finished starting the minipage}%
7015 }
```

End the environment with L^AT_EX processing and closing tag:

```
7016 {%
7017 \LWR@stoppars%
7018 \LWR@endminipage%
  The following empty line is required:
7019
7020 \LWR@htmldivclassend{minipage}%
7021 \LWR@origvspace{1\baselineskip}%
  required for subcaption
7022 \addtocounter{LWR@minipagedepth}{-1}%
7023 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
7024 \global\booltrue{LWR@minipagethispar}%
7025 }
```

71.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output: \parbox [⟨pos⟩] [⟨height⟩] [⟨inner-pos⟩] {⟨width⟩} {⟨text⟩}

A parbox uses the minipage code:

```

7026 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
7027 {
7028 \LWR@traceinfo{parbox of width #4}%
7029 \begin{minipage}[#1][#2][#3]{#4}
7030 #5
7031 \end{minipage}
7032 }

\makebox ((⟨⟩)posn) [(width)] [⟨pos⟩] {⟨text⟩}

7033 \LetLtxMacro{\LWR@origmakebox}\makebox
7034
7035 \RenewDocumentCommand{\makebox}{d() o o m}{%

```

Check for the optional width:

```

7036 \IfValueTF{#2}%
7037 {%

```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```

7038      {%
7039      \def\LWR@align{center}%
7040      \ifstrequal{#3}{l}{\def\LWR@align{left}}{%
7041      \ifstrequal{#3}{r}{\def\LWR@align{right}}{%
7042      \ifstrequal{#3}{s}{\def\LWR@align{justify}}{%

```

To print the width argument:

```

7043      \setlength{\LWR@tempwidth}{#2}%

```

inline-block allows width and text-align to be used in a .

```

7044      \uselengthunit{PT}%
7045      \InlineClass[%
7046          display:inline-block ; %
7047          text-align:\LWR@align\ ; %
7048          width:\rndprintlength{\LWR@tempwidth}%
7049      ]%
7050      {\makebox}%

```

Without a width argument, the text is simply used inline:

```

7051      {\mbox{#4}}%
7052      }% scope
7053 }%

```

```
7054 {\mbox{\#4}}%
7055 }

\framebox [⟨width⟩] [⟨pos⟩] {⟨text⟩}

7056 \LetLtxMacro{\LWR@origframebox}\framebox
7057
7058 \RenewDocumentCommand{\framebox}{o o m}{%
7059 \fbox{\makebox[#1][#2]{#3}}%
7060 }
```

\LWR@forceminwidth {⟨length⟩}

Sets \LWR@tempwidth to be at least 1pt.

```
7061 \newlength{\LWR@atleastonept}
7062
7063 \newcommand*{\LWR@forceminwidth}[1]{%
7064 \setlength{\LWR@atleastonept}{#1}%
7065 \ifthenelse{%
7066     \lengthtest{\LWR@atleastonept>0pt}\AND%
7067     \lengthtest{\LWR@atleastonept<1pt}%
7068 }{%
7069 \setlength{\LWR@atleastonept}{1pt}}%
7070 {}%
7071 }
```

\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.

```
7072 \newcommand*{\LWR@blackborderpadding}{%
7073 \uselengthunit{PT}%
7074 border:\rndprintlength{\LWR@atleastonept} solid black ; %
7075 padding:\rndprintlength{\fboxsep}%
7076 }
```

\fbox {⟨text⟩}

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
7077 \let\LWRprint\fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```

7078 \newcommand{\LWRhtml@fbox}[1]{%
7079 \LWR@traceinfo{HTML fbox}%
7080 \LWR@forceminwidth{\fboxrule}%
7081 \InlineClass[%
7082 \LWR@blackborderpadding%
7083 ]{fbox}{#1}%
7084 }

```

xcolor \lets 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:

```
7085 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

\fboxBlock {<text>} Creates a framed HTML <div> of the text.

A print-output version is also supplied below.

```

7086 \newcommand{\fboxBlock}[1]{%
7087 \LWR@forceminwidth{\fboxrule}%
7088 \begin{BlockClass}[%
7089 \LWR@blackborderpadding%
7090 ]{fboxBlock}%
7091 #1
7092 \end{BlockClass}%
7093 }

```

Env fminipage [<align>] [<height>] [<align>] {<width>}

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```

7094 \NewDocumentEnvironment{fminipage}{O{t} o O{t} m}%
7095 {%
7096 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
7097 \LWR@forceminwidth{\fboxrule}%
7098 \setlength{\LWR@tempwidth}{#4}%
7099 \IfValueTF{#2}{\setlength{\LWR@tempheight}{#2}}{}%
7100 \begin{BlockClass}[%
7101 \LWR@blackborderpadding ; %
7102 \uselengthunit{PT}%
7103 \IfValueTF{#2}{height:\rndprintlength{\LWR@tempheight} ; }{}%
7104 width:\rndprintlength{\LWR@tempwidth}%
7105 ]{fminipage}%
7106 }%
7107 {%
7108 \end{BlockClass}%
7109 \LWR@traceinfo{fminipage done}%

```

```

7110 }

\raisebox {⟨raiselen⟩} [⟨height⟩] [⟨depth⟩] {⟨text⟩}

7111 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
7112
7113 \RenewDocumentCommand{\raisebox}{m o o m}{%
7114 #4%
7115 }

7116 \end{warpHTML}

for HTML & PRINT: 7117 \begin{warpall}

```

`LWRprint@fminipage` is defined inside `warpall`. For print output, it is `\let` to `fminipage`. For `HTML` output, the `HTML` version of `fminipage` is used instead, but the print version is still available for use inside a `lateximage`.

Env `LWRprint@fminipage` [⟨1:align⟩] [⟨2:height⟩] [⟨3:inner-align⟩] {⟨4:width⟩}

Creates a frame around its contents.

```

7118 \newsavebox{\LWR@fminipagebox}
7119
7120 \NewDocumentEnvironment{LWRprint@fminipage}{O{t} o O{t} m}
7121 {%

```

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:

```

7122 \IfValueTF{#3}{%
7123 {\def\LWR@thisalign{#3}}%
7124 {\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.

```

7125 \IfValueTF{#2}{%
7126 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
7127 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%

```

Capture the contents of the environment:

```
7128 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
7129 \IfValueTF{#2}%
7130 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
7131 {\minipage[#1]{#4}}%
7132 }
7133 {%
```

Close the inner minipage and the LR box with the contents:

```
7134 \endminipage%
7135 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
7136 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
7137 \endminipage%
7138 }
```

```
7139 \end{warpall}
```

for PRINT output: 7140 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock {<text>}
```

Creates a framed HTML <div> around the text.

```
7141 \let\fboxBlock\fbox
```

Env fminipage [<align>] [<height>] [<align>] {<width>}

Creates a frame around its contents.

```
7142 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}
7143 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}
```

```
7144 \end{warpprint}
```

72 Direct formatting

\textbf{, etc. are supported, but \bfseries, etc. are not yet supported.}

For high-level block and inline custom CSS classes, see section 39.8.

```
for HTML output: 7145 \begin{warpHTML}

\emph  {\langle text\rangle}

7146 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
7147 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
7148 \LetLtxMacro{\emph}{\LWR@HTMLemph}

\textmd  {\langle text\rangle}

7149 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{\LWR@htmlspanclass{textmd}{#1}}
7150 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
7151 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}

\textbf  {\langle text\rangle}

7152 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
7153 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
7154 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}

\textrm  {\langle text\rangle}

7155 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{\LWR@htmlspanclass{textrm}{#1}}
7156 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
7157 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}

\textsf  {\langle text\rangle}

7158 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{\LWR@htmlspanclass{textsf}{#1}}
7159 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
7160 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}

\texttt  {\langle text\rangle}

7161 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
7162 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
7163 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

```
\textup  {\langle text\rangle}

7164 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{\LWR@htmlspan{#1}{textup}{#1}}
7165 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
7166 \LetLtxMacro{\textup}{\LWR@HTMLtextup}

\textit  {\langle text\rangle}

7167 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}{textit}{#1}}
7168 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7169 \LetLtxMacro{\textit}{\LWR@HTMLtextit}

\textsc  {\langle text\rangle}

7170 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{\LWR@htmlspan{#1}{textsc}{#1}}
7171 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7172 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}

\textsl  {\langle text\rangle}

7173 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{\LWR@htmlspan{#1}{tests1}{#1}}
7174 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7175 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}

\textnormal  {\langle text\rangle}

7176 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7177 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7178 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}

7179 \DeclareRobustCommand{\LWR@nullrmfamily}{}
7180 \DeclareRobustCommand{\LWR@nullsffamily}{}
7181 \DeclareRobustCommand{\LWR@nullttfamily}{}
7182 \DeclareRobustCommand{\LWR@nullbfseries}{}
7183 \DeclareRobustCommand{\LWR@nullmdseries}{}
7184 \DeclareRobustCommand{\LWR@nullupshape}{}
7185 \DeclareRobustCommand{\LWR@nullslshape}{}
7186 \DeclareRobustCommand{\LWR@nullscshape}{}
7187 \DeclareRobustCommand{\LWR@nullitshape}{}
7188 \DeclareRobustCommand{\LWR@nullem}[1]{}
7189 \DeclareRobustCommand{\LWR@nullnormalfont}{}

\LWR@nullfonts  Removes formatting during filename operations.

7190 \newcommand*\LWR@nullfonts{}%
7191 \LetLtxMacro{\emph}{\LWR@nullemph}%
```

```

7192 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7193 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7194 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7195 \LetLtxMacro{\textsf}{\LWR@nulltextsfsf}%
7196 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7197 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7198 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7199 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7200 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7201 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
7202 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7203 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7204 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7205 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7206 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7207 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7208 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7209 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7210 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7211 \LetLtxMacro{\em}{\LWR@nullem}%
7212 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7213 \renewcommand*{\HTMLUnicode}[1]{}%
7214 \renewcommand*{\HTMLEntity}[1]{}%

```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```

7215 \renewcommand*{\&}{and}%
7216 \renewcommand{\textsuperscript}[1]{##1}%
7217 \renewcommand{\textsubscript}[1]{##1}%
7218 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7219 \DeclareExpandableDocumentCommand{\InlineClass}{o m +m}{##3}%
7220 }

```

\mdseries

```
7221 \renewcommand*{\mdseries}{}%
```

\bfseries

```
7222 \renewcommand*{\bfseries}{}%
```

\rmfamily

```
7223 \renewcommand*{\rmfamily}{}%
```

\sffamily

7224 \renewcommand*{\sffamily}{}{}

\ttfamily

7225 \renewcommand*{\ttfamily}{}{}

\upshape

7226 \renewcommand*{\upshape}{}{}

\itshape

7227 \renewcommand*{\itshape}{}{}

\scshape

7228 \renewcommand*{\scshape}{}{}

\normalfont

7229 \renewcommand*{\normalfont}{}{}

\sp {*text*}

For siunitx. Must work in math mode.

7230 \renewcommand{\sp}[1]{\text{^{#1}}}

\sb {*text*}

For siunitx. Must work in math mode.

7231 \renewcommand{\sb}[1]{\text{_{#1}}}

\textsuperscript {*text*}

7232 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}

\textsubscript {*text*}

7233 \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.

7234 \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.

7235 \AtBeginDocument{\let\fup\textsuperscript}

\hfill

7236 \renewcommand*{\hfill}{\quad}

\rulefill

7237 \renewcommand*{\rulefill}{\rule{1in}{1pt}}

\dotfill

7238 \renewcommand*{\dotfill}{\dots}

7239 \end{warpHTML}

73 Skips, spaces, font sizes

for HTML output: 7240 \begin{warpHTML}

\, must be redefined after \RequirePackage{printlen}

7241 \let\LWR@origcomma\,,
 7242 \let\LWR@origtilde~
 7243 \let\LWR@origenskip\enskip
 7244 \let\LWR@origquad\quad
 7245 \let\LWR@origqquad\quad
 7246 \let\LWR@orighspace\hspace
 7247 \let\LWR@origvspace\vspace
 7248 \let\LWR@origrule\rule
 7249 \let\LWR@origmedskip\medskip

Direct-formatting space commands become HTML entities:

```
7250 \renewcommand*{\,}{\HTMLunicode{202f}} % HTML thin non-breakable space

7251
7252 \renewcommand*{-}{\HTMLentity{nbsps}}
7253
7254 \renewcommand*{\textellipsis}{\HTMLunicode{2026}}
```

Direct-formatting font sizes are ignored:

```
7255 \let\LWR@orignormalsize\normalsize
7256 \let\LWR@origsmall\small
7257 \let\LWR@origfootnotesize\footnotesize
7258 \let\LWR@origscriptsize\scriptsize
7259 \let\LWR@origtiny\tiny
7260 \let\LWR@origlarge\large
7261 \let\LWR@origLarge\Large
7262 \let\LWR@origLARGE\LARGE
7263 \let\LWR@orighuge\huge
7264 \let\LWR@origHuge\Huge
7265 \renewcommand*{\normalsize}{}
7266 \renewcommand*{\small}{}
7267 \renewcommand*{\footnotesize}{}
7268 \renewcommand*{\scriptsize}{}
7269 \renewcommand*{\tiny}{}
7270 \renewcommand*{\large}{}
7271 \renewcommand*{\Large}{}
7272 \renewcommand*{\LARGE}{}
7273 \renewcommand*{\huge}{}
7274 \renewcommand*{\Huge}{}
7275
7276 \renewcommand*{\onecolumn}{}
7277
7278 \renewcommand{\twocolumn}[1][]{
7279
7280 #1
7281
7282 }
```

\newline Uses the HTML
 element.

```
7283 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7284 \let\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.

```
7285 \LetLtxMacro{\LWR@origendofline} \\
7286 \NewDocumentCommand{\LWR@endofline}{s o} \\
7287 {\\
7288 \\newline\\
7289 }
```

- \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 <div> 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, l warp correctly suppresses 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. l warp tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around \hspace, \enskip, \quad, and \quadquad until the end of the paragraph, when the closing p tag is created.

When a minipage is seen, the boolean LWR@minipagethispar is set, telling the following horizontal whitespace commands to try to suppress 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 \quadquad's HTML output.

```
7290 \newcommand*{\LWR@minipagestartpars}{%
7291 \ifbool{\LWR@minipagethispar}{\LWR@startpars}{}}%
7292 }
```

- \LWR@minipagestoppars Placed just after \hspace, \quad, or \quadquad's HTML output.

```
7293 \newcommand*{\LWR@minipagestoppars}{%
7294 \ifbool{\LWR@minipagethispar}{\LWR@stoppars}{}}%
7295 }
```

- \quad Handles special minipage & horizontal space interactions.

```
7296 \renewcommand*{\quad}{%
7297 \LWR@minipagestoppars%
7298 \HTMLunicode{2001}%
7299 \LWR@minipagestartpars%
7300 }
```

\qquad Handles special minipage & horizontal space interactions.

```
7301 \renewcommand*{\qquad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```
7302 \renewcommand*{\enskip}{%
7303 \LWR@minipagestoppars%
7304 \HTMLunicode{2000}%
7305 \LWR@minipagestartpars%
7306 }
```

Len \WR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

```
Len \WR@tempheight 7307 \newlength{\LWR@tempwidth}
Len \WR@tempraise 7308 \newlength{\LWR@tempheight}
7309 \newlength{\LWR@tempraise}
```

\LWR@hspace * {*length*}

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.

```
7310 \NewDocumentCommand{\LWR@hspace}{s m}{%
7311 \setlength{\LWR@tempwidth}{#2}%
}
```

If \fill, change to \qquad:

```
7312 \ifnum\gluestretchorder>\LWR@tempwidth>0%
7313 \setlength{\LWR@tempwidth}{2em}%
7314 \fi%
```

Only if the width is not zero:

```
7315 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{0pt}}{}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7316 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
7317 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}{
```

```

7318      {%
7319      \HTMLunicode{2009}% thin breakable space
7320      }%

```

Print the span with the converted width. Not rounded.

```

7321      {%
7322      \uselengthunit{PT}%
7323      \LWR@htmltagc{%
7324      span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
7325      }%
7326      \LWR@htmltagc{/span}%
7327      }%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

7328      \LWR@minipagestartpars%
7329 }% width not 0
7330 }

```

\LWR@nohspase * {*length*}

Used to disable \hspase while creating description \items.

```
7331 \NewDocumentCommand{\LWR@nohspase}{s m}{}
```

\hspase * {*length*}

Handles special minipage & horizontal space interactions.

```
7332 \LetLtxMacro{\hspase}{\LWR@hspase}
```

\LWR@vspace * {*length*} Nullified vspace.

```
7333 \NewDocumentCommand{\LWR@vspace}{s m}{}
```

\vspace * {*length*} Nullified.

```
7334 \let\vspace{\LWR@vspace}
```

\linebreak [*num*] Inserts an HTML br tag.

```
7335 \renewcommand*{\linebreak}[1][]{\newline}
```

\nolinelbreak [*num*]
 7336 \renewcommand*\{\nolinelbreak}[1] [] {}

\pagebreak [*num*] Starts a new paragraph.

7337 \renewcommand*\{\pagebreak}[1] [] {
 7338
 7339 }

\nopagebreak [*num*]
 7340 \renewcommand*\{\nopagebreak}[1] [] {}

\enlargethispage * {*len*}
 7341 \RenewDocumentCommand{\enlargethispage}{s m}{}{}

\LWR@currenttextcolor The color to use for text and \rule, defaulting to black:

7342 \newcommand*\{\LWR@currenttextcolor}{black}

\LWR@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.

7343 \NewDocumentCommand{\LWR@rule}{o m m}{%

The width is copied into a temporary \TeX length, from which comparisons and conversions may be made:

7344 \setlength{\LWR@tempwidth}{#2}%

If it's zero-width then skip the entire rule:

7345 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
 7346 {}% zero-width
 7347 {}% non-zero width

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```

7348      \ifthenelse{%
7349          \lengthtest{\LWR@tempwidth>0pt}\AND%
7350          \lengthtest{\LWR@tempwidth<1pt}%
7351      }%
7352      {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```

7353      \setlength{\LWR@tempheight}{#3}%
7354      \ifthenelse{%
7355          \lengthtest{\LWR@tempheight>0pt}\AND%
7356          \lengthtest{\LWR@tempheight<1pt}%
7357      }%
7358      {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
7359      \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 \TeX code.

```

7360      \uselengthunit{PT}%
7361      \LWR@htmltagc{%
7362      span
7363      style=""%

```

The background color is used to draw the filled rule. The color may be changed by \textcolor .

```
7364      background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```

7365      width:\printlength{\LWR@tempwidth}; %
7366      height:\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 \TeX length allows a typical \TeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \TeX length limits the allowable syntax. To do: A superior method would compute a ratio of \TeX ex height, then print that to HTML with an ex unit.

```

7367      \IfValueTF{#1}%
7368      {%
7369      \setlength{\LWR@tempraise}{0pt-#1}%

```

```

7370      \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7371      \LWR@orignewline%
7372      -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7373      \LWR@orignewline%
7374      -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7375      \LWR@orignewline%
7376      transform: translate(0pt,\printlength{\LWR@tempraise}); %
7377      \LWR@orignewline%
7378  }{}%

```

Display inline-block to place the span inline with the text:

```

7379      display:inline-block;"%
7380  }%
7381  \LWR@htmlltagc{/span}%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

7382      \LWR@minipagestartpars%
7383 }% non-zero width
7384 }

```

`\rule` [$\langle raise \rangle$] $\{ \langle width \rangle \}$ $\{ \langle height \rangle \}$

Handles special minipage & horizontal space interactions.

```
7385 \renewcommand{\rule}{\LWR@rule}
```

```
7386 \end{warpHTML}
```

74 \phantomsection

for HTML output: 7387 `\begin{warpHTML}`

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```

7388 \newcommand*\phantomsection{%
7389 \section*{}%
7390 }

```

```
7391 \end{warpHTML}
```

75 \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>

75.1 HTML logos

for HTML output: 7392 \begin{warpHTML}

\TeX \TeX

latexlogo is a css class used to properly typeset the E and A in \TeX and friends.

latexlogofont is a css class used to select the font for the rest of the logo in \TeX, LuaTeX, ConTeXt, etc.

```
7393 \newcommand*\{\LWR@TeX\}
7394 {\InlineClass{latexlogofont}%
7395 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

\LaTeX \TeX, \TeX_{2ε}

\LaTeXe

```
7396 \newcommand*\{\LWR@LaTeX\}
7397 {\InlineClass{latexlogofont}%
7398 {\InlineClass{latexlogo}%
7399 {L\textsuperscript{a}T\textsubscript{e}X}}}
7400
7401 \renewcommand*\{\LaTeXe\}
7402 {\LaTeX\InlineClass{latexlogofont}%
7403 {\textsubscript{,2}\textsubscript{\textit{\texttt{HTMLunicode{3B5}}}}}}
```

\LuaTeX \LuaTeX, \LuaTeX

\LuaLaTeX

```
7404 \newcommand*\{\LWR@LuaTeX\}{\InlineClass{latexlogofont}{Lua}\TeX}
7405 \newcommand*\{\LWR@LuaLaTeX\}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

\XeTeX X_ET_EX, X_ET_EX
 \XeLaTeX xetexlogo is a css class which aligns the backwards E in X_ET_EX and spaces T_EX appropriately.

xelatexlogo is a css class which aligns the backwards E in X_ET_EX and spaces T_EX appropriately.

```
7406 \newcommand*{\Xe}{%
 7407   {\X\textsubscript{\HTMLUnicode{18e}}}}%
 7408 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}%
 7409 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

\ConTeXt ConTeXt

```
7410 \newcommand*{\LWR@ConTeXt}{%
 7411   {\InlineClass{latexlogofont}{Con}\TeX{}}%
 7412 \InlineClass{latexlogofont}{t}}
```

\BibTeX BIBT_EX, *MakeIndex*
 \MakeIndex
 7413 \providecommand*{\BibTeX}{%
 7414 {\InlineClass{latexlogofont}{B\textsc{ib}}}\TeX}%
 7415
 7416 \newcommand*{\MakeIndex}{%
 7417 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}}

\AmS A_MS

amslogo is a css class used for the A_MS logo.

```
7418 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
 7419   {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

\MiKTeX MiKT_EX

```
7420 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

\LyX LyX

lyxlogo is a css class used for the LyX logo.

```
7421 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}%
 7422 \end{warpHTML}
```

75.2 Print logos

```
for PRINT output: 7423 \begin{warpprint}
7424 \newcommand*{\XeTeXrevE}{%
7425   {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}}
7426 \providecommand*{\XeTeX}{\mbox{X}\XeTeXrevE\TeX}
7427 \providecommand*{\XeLaTeX}{\mbox{X}\XeTeXrevE\LaTeX}
7428 \providecommand*{\AmS}{%
7429 \leavevmode\hbox{$\mathcal A$\kern-.2em\lower.376ex%}
7430 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
7431 \newcommand*{\LyX}{\textsf{LyX}}
7432 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
7433 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
7434 \providecommand*{\BibTeX}{\mbox{B}\textit{texsc}{ib}\TeX}
7435 \providecommand*{\MakeIndex}{\mbox{\textit{texit}{MakeIndex}}}
7436 \providecommand*{\ConTeXt}{\mbox{Con\TeX{}t}}
7437 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
7438 \end{warpprint}
```

76 \AtBeginDocument, \AtEndDocument

```
for HTML output: 7439 \begin{warpHTML}
```

```
\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the document.
\LWR@LwarpEnd
7440 \AfterEndPreamble{\LWR@LwarpStart}
7441 \AtEndDocument{\LWR@LwarpEnd}

7442 \end{warpHTML}
```

77 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 **lwarf-a4.sty**

§ 78 Package **a4**

Pkg a4 a4 is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4}

2 \newcommand*{\WideMargins}{}

File 3 **lwarf-a4wide.sty**

§ 79 Package **a4wide**

Pkg a4wide a4wide is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4wide}

File 4 **lwarf-a5comb.sty**

§ 80 Package **a5comb**

Pkg a5comb a5comb is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a5comb}

File 5 **lwarf-abstract.sty**

§ 81 Package **abstract**

(Based on original code by PETER WILSON.)

Pkg abstract abstract is supported and patched by lwarf.

 **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: Accept all options for \warp-abstract:

```
1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\@bsrunintitle}{%
11 \hspace*{\abstitleskip}%
12 {\abstractnamefont{%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }%
16
17 \if@titlepage
18 \ renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \begin{parpenalty}\@lowpenalty
22 \if@bsrunin
23 \else
24 \if@bsstyle
25 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}%
26 \else
27 \ifnumber@bs
28 \num@bs
29 \else
30 \begin{absnamepos}%
31 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}%
32 \end{parpenalty}\@M
33 \end{absnamepos}%
34 %% \vspace{\abstitleskip}%
35 \fi
36 \fi
37 \vspace{\abstitleskip}%
38 \fi
39 \put@bsintoc%
40 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42 }
43 \else
44 \renewenvironment{abstract}{%
```

```

45      \if@bsrunin
46      \else
47          \if@bsstyle
48              \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49          \else
50              \ifnumber@bs
51                  \num@bs
52              \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %%           \vspace{\abstitleskip}%
57           \fi
58           \fi
59           \vspace{\abstitleskip}%
60           \fi
61           \put@bsintoc%
62           \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63           {\par\end{@bstr@ctlist}}
64 \fi
65

```

File 6 **lwarf-addlines.sty**

§ 82 Package **addlines**

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 7 **lwarf-afterpage.sty**

§ 83 Package **afterpage**

Pkg **afterpage** Emulated.

for HTML output: Discard all options for lwarf-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```

File 8 **lwarp-algorithmicx.sty**

§ 84 Package **algorithmicx**

Pkg **algorithmicx** algorithmicx is supported with minor adjustments.

for HTML output: 1 \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 221.1.

for HTML output: 2 \begin{warpHTML}

```
3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*\ALG@doentity{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmntagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmntagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}{}}%
19 }
```

20 \end{warpHTML}

File 9 **lwarp-alltt.sty**

§ 85 Package **alltt**

Pkg **alltt** alltt is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{alltt}

```

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
7 }
8 \AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
9 }

```

File 10 **l warp-amsthm.sty**

§ 86 Package **amsthm**

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by l warp.

CSS styling of theorems and proofs:

Theorem: <div> of class amsthmbody<theoremstyle>
Theorem Name: of class amsthmname<theoremstyle>
Theorem Number: of class amsthmnumber<theoremstyle>
Theorem Note: of class amsthmnote<theoremstyle>
Proof: <div> of class amsthmproof
Proof Name: of class amsthmproofname
where <theoremstyle> is plain, definition, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

3 \renewcommand{\theoremstyle}[1]{%
4   \@ifundefined{th@\#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}%
8   }{%
9     \thm@style{#1}%
}

```

```

10      \renewcommand{\LWR@newtheoremstyle}{#1}%
11  }%
12 }

```

Patched to remember the style for this theorem type:

```

13 \def\@xnthm#1#2{%
14   \csedef{\LWR@thmstyle#2}{\LWR@newtheoremstyle}%
15   \let\@tempa\relax
16   \xp@ifdefinable\csname #2\endcsname{%
17     \global\xp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\xp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\xp\@nx\csname th@\the\thm@style\endcsname}%
22             {}{##1}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\oparg{\cynthm{#2}}[]}%
25       \fi
26     \AtBeginEnvironment{#2}{\edef{\LWR@thisthmstyle}{\csuse{\LWR@thmstyle#2}}}%
27   }%
28   \atempa
29 }

```

Patched to enclose with css:

```

30 \newcommand{\LWR@haveamsthmname}{%
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname}\LWR@thisthmstyle{##1}}
32 }%
33
34 \newcommand{\LWR@haveamsthmnumber}{%
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber}\LWR@thisthmstyle{##1}}
36 }%
37
38 \newcommand{\LWR@haveamsthmnote}{%
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote}\LWR@thisthmstyle{##1}}
40 }%
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for css:

```

45 \def\@begintheorem#1#2[#3]{%
46 \LWR@forcenewpage% new
47   \BlockClass{amsthmbody}\LWR@thisthmstyle}% new
48   \deferred@thm@head{%
49     \the\thm@headfont \thm@indent

```

```

50  \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}\% new
51  \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}\% new
52  \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}\% new
53  \thm@swap\swappedhead\thmhead{#1}{#2}{#3}\%
54  \the\thm@headpunct-
55  \thmheadnl % possibly a newline.
56  \hskip\thm@headsep
57 }\%
58 \ignorespaces

```

Patched for css:

```
59 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }
```

Proof QED symbol:

```

60 \AtBeginDocument{
61 \def\openbox{\text{\HTMLunicode{25A1}}}\% UTF-8 white box
62 \def\blacksquare{\text{\HTMLunicode{220E}}}\% UTF-8 end-of-proof
63 \def\Box{\text{\HTMLunicode{25A1}}}\% UTF-8 white box
64 }

```

Patched for css:

```

65 \renewenvironment{proof}[1][\proofname]{\par
66 \LWR@forcenewpage\% new
67   \BlockClass{amsthmproof}\% new
68   \pushQED{\qed}\%
69   \normalfont \topsep6\p@\oplus6\p@\relax
70   \trivlist
71   \item[\hskip\labelsep
72     \InlineClass{amsthmproofname}{\#1\@addpunct{.}}]\ignorespaces\% changes
73 }\%
74 \InlineClass{theoremendmark}{\popQED}\endtrivlist%
75 \endBlockClass\% new
76 \@endpefalse
77 }

```

File 11 **l warp-any size.sty**

§ 87 Package **any size**

Pkg **any size** any size is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{any size}

```
2 \def\papersize#1#2{}
3 \def\marginsize#1#2#3#4{}
```

File 12 l warp-appendix.sty

§ 88 Package **appendix**

Pkg appendix appendix is patched for use by l warp.

- ⚠ 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:

```
1 \LWR@ProvidesPackagePass{appendix}

2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addapptocentry
6 \fi
7 }
8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addapptocentry
13 \fi
14 }
```

File 13 l warp-arabicfront.sty

§ 89 Package **arabicfront**

Pkg arabicfront arabicfront is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{arabicfront}
```

File 14 l warp-authblk.sty

§ 90 Package **authblk**

Pkg authblk authblk is patched for HTML.

package support l warp supports the native L^AT_EX 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 52.7.

(Based on original code by PATRICK W. DALY.)

for HTML output: Require that authblk be loaded before titling:

```
1 \@ifpackageloaded{titling}{  
2 \PackageError{l warp-authblk}{  
3 {Package authblk must be loaded before titling}  
4 {Titling appends authblk's author macro, so authblk must be loaded first.}  
5 }  
6 }
```

Load authblk:

```
7 \LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```
8 \LetLtxMacro{\LWR@affil}{\affil}  
9  
10 \renewcommand{\affil}[2][]{%  
11 \LWR@affil[#1]{\protect\InlineClass{affiliation}{#2}}  
12 }
```

Create an HTML break for an \authorcr:

```
13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

File 15 l warp-balance.sty

§ 91 Package **balance**

Pkg balance Emulated.

for HTML output: Discard all options for l warp-balance:

```
1 \LWR@ProvidesPackageDrop{balance}  
  
2 \newcommand*{\balance}{}  
3 \newcommand*{\nobalance}{}  

```

File 16 **lwarf-bigdelim.sty**

§92 Package **bigdelim**

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 `` class of `ldelim` or `rdelim`, and the default css sets this to `font-size:200%`

for HTML output: 1 `\LWR@ProvidesPackagePass{bigdelim}`

Remember the print-mode versions:

2 `\LetLtxMacro{\LWR@origldelim}{\ldelim}`
3 `\LetLtxMacro{\LWR@origrdelim}{\rdelim}`

4 `\RenewDocumentCommand{\ldelim}{m m m O{} }{%`
5 `\renewcommand{\LWR@multirowborder}{right} %`
6 `\multirow{#2}{#3}{#4} \InlineClass{ldelim}{#1} %`
7 `}`
8
9 `\RenewDocumentCommand{\rdelim}{m m m O{} }{%`
10 `\renewcommand{\LWR@multirowborder}{left} %`
11 `\multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4} %`
12 `}`

When entering a `lateximage`, restore the print-mode versions:

13 `\appto{\LWR@restoreorigformatting}{%`
14 `\LetLtxMacro{\ldelim}{\LWR@origldelim} %`
15 `\LetLtxMacro{\rdelim}{\LWR@origrdelim} %`
16 `}`

File 17 **lwarf-bigstrut.sty**

§93 Package **bigstrut**

Pkg **bigstrut** `bigstrut` is used as-is for print or `lateximage`, and patched for HTML.

for HTML output:

```

1 \LWR@ProvidesPackagePass{bigstrut}

2 \LetLtxMacro{\LWR@origbigstrut}{\bigstrut}
3
4 \renewcommand{\bigstrut}[1][x]{}
5
6 \appto{\LWR@restoreorigformatting}{%
7 \LetLtxMacro{\bigstrut}{\LWR@origbigstrut}%
8 }
```

File 18 **l warp-bookmark.sty**

§ 94 Package **bookmark**

Pkg **bookmark** bookmark is emulated during HTML output, and the bookmark package is ignored.

for HTML output: Discard all options for l warp-bookmark:

```

1 \LWR@ProvidesPackageDrop{bookmark}

2 \newcommand*{\bookmarksetup}[1]{}
3 \newcommand*{\bookmarksetupnext}[1]{}
4 \newcommand*{\bookmark}[2][]{}
5 \newcommand*{\bookmarkdefinestyle}[2]{}
6 \newcommand*{\bookmarkget}[1]{}
7 \newcommand{\BookmarkAtEnd}[1]{}
```

File 19 **l warp-booktabs.sty**

§ 95 Package **booktabs**

Pkg **booktabs** booktabs is emulated during HTML output, and used as-is during print output and inside an HTML `latextimage`.

for HTML output:

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML `latextimages`. The HTML versions temporarily overwrite these print versions when `tabular` is started.

```

2 \LetLtxMacro{\LWR@origtoprule}{\toprule}
3 \LetLtxMacro{\LWR@origmidrule}{\midrule}
4 \LetLtxMacro{\LWR@origcmidrule}{\cmidrule}
```

```
5 \LetLtxMacro{\LWR@origbottomrule}{\bottomrule}
6 \LetLtxMacro{\LWR@origaddlinespace}{\addlinespace}
7 \LetLtxMacro{\LWR@origmorecmidrules}{\morecmidrules}
8 \LetLtxMacro{\LWR@origspecialrule}{\specialrule}
```

File 20 **l warp-boxedminipage.sty**

§ 96 Package **boxedminipage**

Pkg **boxedminipage** **boxedminipage** is superceded by **boxedminipage2e**.

for HTML output: 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

File 21 **l warp-boxedminipage2e.sty**

§ 97 Package **boxedminipage2e**

Pkg **boxedminipage2e** **boxedminipage2e** is emulated.

for HTML output: Discard all options for **l warp-boxedminipage2e**:

```
1 \LWR@ProvidesPackageDrop{boxedminipage2e}

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}%
4 \minipage{%
5 }%
6 {%
7 \endminipage{%
8 \end{BlockClass}%
9 }}
```

File 22 **l warp-caption2.sty**

§ 98 Package **caption2**

Pkg **caption2** **caption2** is not used. The user is recommended to use **caption** instead.

for HTML output: 1 \LWR@loadnever{caption2}{caption}

File 23 **lwarf-ccaption.sty**

§ 99 Package **ccaption**

Pkg ccaption ccaption is not used. The user is recommended to use caption instead.

for HTML output: 1 \LWR@loadnever{ccaption}{caption}

File 24 **lwarf-changepage.sty**

§ 100 Package **changepage**

Pkg changepage changepage is ignored.

for HTML output: Discard all options for lwarf-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changeppage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}
```

File 25 **lwarf-chngpage.sty**

§ 101 Package **chngpage**

Pkg chngpage chngpage is superceded by changepage.

for HTML output: 1 \LWR@loadnever{chngpage}{changepage}

File 26 **lwarf-chappg.sty**

§ 102 Package **chappg**

Pkg chappg chappg is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{chappg}
2 \renewcommand{\pagenumbering}[2][]{}
3 \providecommand{\chappgsep}{--}
```

File 27 **l warp-color.sty**

§ 103 Package **color**

Pkg **color** Allowed but ignored. **xcolor** is then required as well.

color is superceded by **xcolor**, and **l warp** 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:

```
1 \LWR@ProvidesPackagePass{color}
2 \RequirePackage{xcolor}
```

File 28 **l warp-crop.sty**

§ 104 Package **crop**

Pkg **crop** Emulated.

for HTML output: Discard all options for **l warp-crop**:

```
1 \LWR@ProvidesPackageDrop{crop}
2 \newcommand*\crop[1][]{}
3 \newcommand*\cropdef[6][]{}
```

File 29 **l warp-cutwin.sty**

§ 105 Package **cutwin**

Pkg **cutwin** Emulated.

for HTML output: Discard all options for **l warp-cutwin**:

```
1 \LWR@ProvidesPackageDrop{cutwin}
```

```

2 \newcommand*{\opencutleft}{}
3 \newcommand*{\opencutright}{}
4 \newcommand*{\opencutcenter}{}
5 \newcommand*{\cutfuzz}{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{}
12
13 \newcommand*{\pageinwindow}{%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}
17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*{\putstuffinpic}{}
24
25 \newcommand*{\picinwindow}{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}

```

File 30 **l warp-dcolumn.sty**

§ 106 Package **dcolumn**

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```
1 \LWR@ProvidesPackageDrop{dcolumn}
```

File 31 **l warp-draftwatermark.sty**

§ 107 Package **draftwatermark**

Pkg draftwatermark draftwatermark is emulated during HTML output, and the draftwatermark package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{draftwatermark}

```

2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}

```

File 32 **lwarf-ebook.sty**§ 108 Package **ebook**

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 33 **lwarf-ellipsis.sty**§ 109 Package **ellipsis**

Pkg **ellipsis** ellipsis is emulated during HTML output, and the ellipsis package is ignored.

```

1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}

```

File 34 **lwarf-emptypage.sty**§ 110 Package **emptypage**

Pkg **emptypage** emptypage is ignored.

for HTML output: Discard all options for l warp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

File 35 **l warp-endnotes.sty**

§ 111 Package **endnotes**

(Based on original 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: 1 \LWR@ProvidesPackagePass{endnotes}

```
2 \def\enoteformat{%
3   \rightskip\z@ \leftskip\z@ \parindent=1.8em
4   \leavevmode
5   \llap{
6     \makeenmark
7   }
8 }
9
10 \def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}
```

File 36 **l warp-enumerate.sty**

§ 112 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 lwarf and thus must no longer be dropped.

for HTML output: 1 \LWR@ProvidesPackagePass{enumerate}

File 37 **lwarf-enumitem.sty**

§ 113 Package **enumitem**

Pkg enumitem enumitem is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{enumitem}

for HTML output: 2 \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.

```
3 \let\LWR@orignewlist\newlist
4
5 \renewcommand*{\newlist}[3]{%
6 \LWR@orignewlist{#1}{#2}{#3}%
7 \AtBeginEnvironment{#1}{\csuse{\LWR@#2start}}%
8 \AtEndEnvironment{#1}{\csuse{\LWR@#2end}}%
9 }
```

10 \end{warpHTML}

File 38 **lwarf-epigraph.sty**

§ 114 Package **epigraph**

Pkg epigraph epigraph is emulated during HTML output, and the epigraph package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

```
2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
```

```

5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}

```

Use css to format epigraphs.

The following are null commands for source compatibility:

```

21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{\flushepinormal}{}{}
24 \newcommand{\textflush}[1]{\flushepinormal}
25 \newcommand{\epigraphflush}[1]{\flushright}
26 \newcommand{\sourceflush}[1]{\flushright}
27 \newcommand*{\epigraphsize}{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}
31 \newcommand{\epigraphhead}[2][0]{#2}
32 \newcommand{\dropchapter}[1]{}
33 \newcommand*{\undodrop}{}
34 \newcommand{\cleartoevenpage}[1]{}

```

File 39 **lwarp-eso-pic.sty**

§ 115 Package **eso-pic**

Pkg **eso-pic** **eso-pic** is emulated during HTML output, and the **eso-pic** package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{eso-pic}

2 \newcommand*{\LenToUnit}{}

```

```

3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \renewcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*\ClearShipoutPictureBG(){}
16 \newcommand*\ClearShipoutPicture(){}
17 \newcommand*\ClearShipoutPictureFG(){}
18 \newcommand{\gridSetup}[6][]{}

```

File 40 **lwarf-everypage.sty**

§ 116 Package **everypage**

Pkg **everypage** everypage is emulated during HTML output, and the everypage package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{everypage}

2 \newcommand*\AddEverypageHook[1]{}
3 \newcommand*\AddThispageHook[1]{}

```

File 41 **lwarf-everyshi.sty**

§ 117 Package **everyshi**

Pkg **everyshi** Emulated.

for HTML output: Discard all options for lwarf-everyshi:

```

1 \LWR@ProvidesPackageDrop{everyshi}

2 \newcommand*\EveryShipout[1]{}
3 \newcommand*\AtNextShipout[1]{}

```

File 42 **l warp-extramarks.sty**

§ 118 Package **extramarks**

Pkg **extramarks** extramarks is not used.

for HTML output: Discard all options for l warp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}

2 \newcommand*{\extramarks}[2]{}
3 \newcommand*{\firstleftxmark}{}
4 \newcommand*{\lastleftxmark}{}
5 \newcommand*{\firstrightxmark}{}
6 \newcommand*{\lastrightxmark}{}
7 \newcommand*{\firstxmark}{}
8 \newcommand*{\lastxmark}{}
9 \newcommand*{\topxmark}{}
10 \newcommand*{\topleftxmark}{}
11 \newcommand*{\firstleftmark}{}
12 \newcommand*{\lastrightmark}{}
```

File 43 **l warp-fancybox.sty**

§ 119 Package **fancybox**

(Based on original 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 \teximage, which is done by adding \teximage at the very start of FramedEqn's beginning code, and \endteximage at the very end of the ending code. Unfortunately, the HTML alt attribute is not used here.

```
\newenvironment{FramedEqn}
{
  \latextimage% NEW
  \setlength{\fboxsep}{15pt}
  ...}{...
  \[\fbox{\TheSbox}\]
  \endlatextimage% 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}
  \begin{Verbatim}
  \end{Verbatim}
  \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.

for HTML output: 1 `\begin{warpHTML}`

```

2 \LWR@ProvidesPackagePass{fancybox}

3 \renewcommand*{\@shadowbox}[1]{\InlineClass{shadowbox}{#1}}
4
5 \renewcommand*{\@doublebox}[1]{\InlineClass{doublebox}{#1}}
6
7 \renewcommand*{\@ovalbox}[2]{%
8 \ifthenelse{\isequivalentto{#1}{\thinlines}}
9 {\InlineClass{ovalbox}{#2}}
10 {\InlineClass{Ovalbox}{#2}}
11 }

```

Convert minipages, parboxes, and lists into linear text using the `\LWR@nestspan` environment:

```

12 \let\LWR@origSbox\Sbox
13
14 \def\Sbox{\LWR@origSbox\LWR@nestspan}
15
16
17 \let\LWR@origendSbox\endSbox
18
19 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

`\begin{Beqnarray}` is adapted for MathJax or enclosed inside a `\teximage`:

```

20 \RenewEnviron{Beqnarray}
21 {\LWR@eqnarrayfactor}
22
23 \csgpreto{Beqnarray}{\boolfalse{\LWR@numbereqnarray}}

```

`\GenericCaption` is enclosed in an HTML block:

```

24 \renewcommand{\GenericCaption}[1]{%
25 \LWR@htmlblocktag{figcaption}%
26 #1%
27 \LWR@htmlblocktag{/figcaption}%
28 }

```

`\Btrivlist` is enclosed in an HTML block:

```

29 \RenewDocumentEnvironment{Btrivlist}{m o}
30 {\begin{BlockClass}{Btrivlist}\tabular{#1}}
31 {\endtabular\end{BlockClass}}

```

`\Btrivlist` is also neutralized when used inside a span:

```

32 \AtBeginEnvironment{\LWR@nestspan}{%
33 \RenewDocumentEnvironment{Btrivlist}{m o}{}{}%
34 \RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
35 }

```

lwarp's handling of `\item` is patched to accept fancybox's optional arguments:

```

36 \let\LWRFB@origitemizeitem\LWR@itemizeitem
37 \let\LWRFB@origdescitem\LWR@descitem
38 \LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}
39
40 \RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
41 \RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}

```

The various boxed lists become regular lists:

```

42 \renewenvironment{Bitemize}[1][]{\begin{itemize}}{\end{itemize}}
43 \renewenvironment{Benumerate}[1][]{\begin{enumerate}}{\end{enumerate}}
44 \renewenvironment{Bdescription}[1][]{\begin{description}}{\end{description}}

```

`\boxput` simply prints one then the other argument, side-by-side instead of above and behind:

```

45 \RenewDocumentCommand{\boxput}{s d() m m}{%
46 \IfBooleanTF{#1}{\quad\#4}{\quad\#3}\quad\#3}%
47 }

```

Neutralized commands:

```

48 \RenewDocumentCommand{\fancyput}{s d() m m}{}
49 \RenewDocumentCommand{\thisfancyput}{s d() m m}{}
50
51 \RenewDocumentCommand{\fancypage}{m m m m}{}
52 \RenewDocumentCommand{\thisfancypage}{m m m m}{}
53
54 \def\LandScape#1{}
55 \def\endLandScape{}
56 \def\@Landscape#1#2#3{}
57 \def\endLandscape{}

```

Low-level patches for Verbatim, VerbatimInput, UseVerbatim:

```

58 \let\LWRFB@UseVerbatim\UseVerbatim
59 \renewcommand*{\UseVerbatim}[1]{%
60 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-5\baselineskip}%%
61 \LWRFB@UseVerbatim{#1}%
62 \LWR@afterendverbatim%

```

```

63 }
64
65 \let\LWRFB@LUseVerbatim\LUseVerbatim
66
67 \renewcommand*{\LUseVerbatim}[1]{%
68 \LWR@atbeginverbatim{LVerbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
69 \noindent%
70 \LWRFB@LUseVerbatim{#1}%
71 \LWR@afterendverbatim%
72 }
73
74 \def\@BUseVerbatim[#1]#2{%
75 \LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{- .5\baselineskip}%%
76 \LWRFB@UseVerbatim{#2}%
77 \LWR@afterendverbatim%
78 }

79 \end{warpHTML}

```

File 44 **l warp-fancyhdr.sty**

§ 120 Package **fancyhdr**

Pkg **fancyhdr** fancyhdr is nullified.

for HTML output: Discard all options for l warp-fancyhdr:

```

1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*{\fancyhead}[2][]{}
3 \newcommand*{\fancyfoot}[2][]{}
4 \newcommand*{\fancyhf}[2][]{}
5 \newcommand*{\fancypagestyle}[2] {}
6 \newcommand*{\lhead}[2][]{}
7 \newcommand*{\chead}[2][]{}
8 \newcommand*{\rhead}[2][]{}
9 \newcommand*{\lfoot}[2][]{}
10 \newcommand*{\cfoot}[2][]{}
11 \newcommand*{\rfoot}[2][]{}
12 \newcommand*{\headrulewidth}{}}
13 \newcommand*{\footrulewidth}{}}
14 \newcommand*{\fancyheadoffset}[2][]{}
15 \newcommand*{\fancyfootoffset}[2][]{}
16 \newcommand*{\fancyhfoffset}[2][]{}
17 \newcommand*{\iffloatpage}[2]{#2}
18 \newcommand*{\ifftopfloat}[2]{#2}
19 \newcommand*{\iffbotfloat}[2]{#2}

```

File 45 **l warp-fancyvrb.sty**

§ 121 Package **fancyvrb**

(Based on original code by TIMOTHY VAN ZANDT.)

Pkg fancyvrb fancyvrb is supported with some patches.

for HTML output: 1 \RequirePackage{xcolor}%; for \convertcolorspec
2 \LWR@ProvidesPackagePass{fancyvrb}

for HTML output: 3 \begin{warpHTML}

Initial default patch for fancyvrb:

4 \fvset{frame=none}%;

For \VerbatimFootnotes:

5 \renewcommand{\VerbatimFootnotes}{
6 \PackageError{l warp}{
7 {Verbatim footnotes are not yet supported by l warp.}
8 {This may be improved some day.}
9 }}

After the preamble is loaded, after any patches to Verbatim:

10 \AfterEndPreamble{
11 \LWR@traceinfo{Patching Verbatim.}}

Remember the original defintion of Verbatim:

12 \let\LWRFV@origVerbatim\Verbatim

Env Verbatim Patched to place the environment in a fancyvrb div, and the label in a fancyvrblabel div. Also corrects the left margin for line numbers. Also uses VerbatimHTMLWidth to control placement of line numbers on the right. Aligning the right margin requires knowing the width.

13 \renewcommand*{\Verbatim}{%
14 \LWR@forcenewpage
15 \LWRFV@origVerbatim%
16 }

\LWR@FVstyle Holds the style of the verbatim.

```
17 \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.

```
18 \newcommand*{\LWR@fvstartnone}{%
19 \LWR@traceinfo{fvstartnone}%
20 \hbox to\z@\{\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}\}%
21 }%
22
23 \newcommand*{\LWR@fvendnone}{%
24 \LWR@traceinfo{fvendnone}%
25 \hbox to\z@\{\LWR@afterendverbatim\}%
26 }%
27
28 \newcommand*{\LWR@fvstartsingle}{%
29 \LWR@traceinfo{fvstartsingle}%
30 \LWR@fvstartnone%
31 \FV@BeginListFrame@Single%
32 }%
33
34 \newcommand*{\LWR@fvendsingle}{%
35 \LWR@traceinfo{fvendsingle}%
36 \FV@EndListFrame@Single%
37 \LWR@fvendnone%
38 }%
39
40 \newcommand*{\LWR@fvstartline}{%
41 \LWR@traceinfo{fvstartline}%
42 \LWR@fvstartnone%
43 \FV@BeginListFrame@Lines%
44 }%
45
46 \newcommand*{\LWR@fvendline}{%
47 \LWR@traceinfo{fvendline}%
48 \FV@EndListFrame@Lines%
49 \LWR@fvendnone%
50 }%
```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```
51
52 \newcommand*{\LWR@FVfindcurrentcolor}{%
53 \protect\colorlet{\LWR@current@color}{.}%
54 \protect\convertcolorspec[named]{\LWR@current@color}{HTML}\LWR@tempcolor%
```

```
55 }
56
57 \newcommand*{\LWR@FVtextstyle}{%
58 \LWR@FVfindcurrentcolor
59 \ifdefstring{\LWR@tempcolor}{000000}%
60 {}%
61 {color: \#\LWR@tempcolor ; }%
62 }
63
64
65 \newcommand*{\LWR@FVfindbordercolor}{%
66 \FancyVerbRuleColor%
67 \LWR@FVfindcurrentcolor
68 \color{black}
69 }
70
71 % border width of \FV@FrameRule
72 \newcommand*{\LWR@FVborderstyle}[1]{%
73 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ;
74 \LWR@FVfindbordercolor
75 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ;
76 }
77
78 \def\FV@Frame@none{%
79 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle}%
80 \let\FV@BeginListFrame\LWR@fvstartnone%
81 \let\FV@LeftListFrame\relax%
82 \let\FV@RightListFrame\relax%
83 \let\FV@EndListFrame\LWR@fvendnone}
84
85 \FV@Frame@none% default values
86
87 \def\FV@Frame@singl{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{} }%
89 \let\FV@BeginListFrame\LWR@fvstartsingl%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
91 \let\FV@RightListFrame\FV@RightListFrame@Single%
92 \let\FV@EndListFrame\LWR@fvendsingl}
93
94 \def\FV@Frame@lines{%
95 \renewcommand*{\LWR@FVstyle}{%
96     \LWR@FVtextstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
97 }%
98 \let\FV@BeginListFrame\LWR@fvstartline%
99 \let\FV@LeftListFrame\relax%
100 \let\FV@RightListFrame\relax%
101 \let\FV@EndListFrame\LWR@fvendline}
102
103 \def\FV@Frame@topline{%
104 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}}%
```

```

105 \let\FV@BeginListFrame\LWR@fvstartline%
106 \let\FV@LeftListFrame\relax%
107 \let\FV@RightListFrame\relax%
108 \let\FV@EndListFrame\LWR@fvendnone}
109
110 \def\FV@Frame@bottomline{%
111 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-bottom}}%
112 \let\FV@BeginListFrame\LWR@fvstartnone%
113 \let\FV@LeftListFrame\relax%
114 \let\FV@RightListFrame\relax%
115 \let\FV@EndListFrame\LWR@fvendline}
116
117 \def\FV@Frame@leftline{%
118 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-left}}%
119 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
120 \ifx\FancyVerbFillColor\relax%
121 \let\FV@FrameFillLine\relax%
122 \else%
123 \tempdima\FrameRule\relax%
124 \multiply\tempdima-\tw@%
125 \edef\FV@FrameFillLine{%
126 {\noexpand\FancyVerbFillColor{\vrule\width\number\tempdima sp}%
127 \kern-\number\tempdima sp}}%
128 \fi%
129 \let\FV@BeginListFrame\LWR@fvstartnone%
130 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
131 \let\FV@RightListFrame\relax%
132 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

133 \def\FV@SingleFrameLine#1{%
134   \hbox to\z@{%
135     \kern\leftmargin
136     \ifnum#1=\z@\relax
137       \let\FV@Label\FV@LabelBegin
138     \else
139       \let\FV@Label\FV@LabelEnd
140     \fi
141     \ifx\FV@Label\relax
142       \FancyVerbRuleColor{\vrule \width\linewidth \height\FrameRule}%
143     \else
144       \ifnum#1=\z@
145         \setbox\z@\hbox{\strut\enspace\LabelBegin\enspace\strut}%
146         \ifx\LabelPositionTopLine\relax
147           \else
148             \LWR@FVfindbordercolor
149             \LWR@htmllagc{div class="fancyvrblabel" style="color: \tempcolor"}%

```

```

150      \LWR@origtextrm{\FV@LabelBegin}%
151      \LWR@htmllagc{/div}
152      \fi
153  \else
154 %       \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
155       \ifx\FV@LabelPositionBottomLine\relax
156       \else
157       \LWR@FVfindbordercolor
158       \LWR@htmllagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
159       \LWR@origtextrm{\FV@LabelEnd}
160       \LWR@htmllagc{/div}
161       \fi
162   \fi
163
164   \fi
165   \hss
166 }
167 }
```

Processes each line, adding optional line numbers. Original code is from the `fancyvrb` package.

```

168 \def\FV@ListProcessLine#1{%
169   \hbox to \hsize{%
170 %     \kern\leftmargin
171     \hbox to \VerbatimHTMLWidth {%
172       \ifcsvvoid{\FV@LeftListNumber}{}{\kern 2.5em}%
173       \FV@LeftListNumber%
174 %       \FV@LeftListFrame
175       \FancyVerbFormatLine{#1}%
176       \hss%
177 %       \FV@RightListFrame
178       \FV@RightListNumber%
179     }%
180     \hss% required to avoid underfull hboxes
181 }
182 }
```

Env `BVerbatim`

```

183 \AtBeginEnvironment{BVerbatim}
184 {
185 \LWR@forcenewpage
186 \LWR@atbeginverbatim{bverbatim}
187
188 }
189
190 \AfterEndEnvironment{BVerbatim}
191 {
```

```
192 \leavevmode\par\LWR@origvspace{-\baselineskip}
193 \LWR@afterendverbatim
194 }
```

End of the modifications to make at the end of the preamble:

```
195 } % \AfterEndPreamble
196 \end{warpHTML}
```

File 46 **l warp-figcaps.sty**

§ 122 Package **figcaps**

Pkg **figcaps** Emulated.

for HTML output: Discard all options for l warp-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 47 **l warp-fix2col.sty**

§ 123 Package **fix2col**

Pkg **fix2col** fix2col is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fix2col}

File 48 **l warp-float.sty**

§ 124 Package **float and \newfloat**

Pkg float float is emulated during HTML output, and the float package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{float}[2016/03/04]

See section 61.2 for the \listof command.

\newfloat {\langle 1: type\rangle} {\langle 2: placement\rangle} {\langle 3: ext\rangle} [{\langle 4: within\rangle}]

Emulates the \newfloat command from the float package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}{%
4 {\\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}%
5 {\\DeclareFloatingEnvironment[fileext=#3]{#1}}}
```

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.

```
6 \cslet{listof#1s}\relax
7 \cslet{listof#1es}\relax
8 }
```

\floatname {\langle type\rangle} {\langle name\rangle}

Sets the text name of the float, such as “Figure”.

```
9 \NewDocumentCommand{\floatname}{m +m}{%
10 \SetupFloatingEnvironment{#1}{name=#2}%
11 }
```

\floatplacement {\langle type\rangle} {\langle placement\rangle}

Float placement is ignored.

```
12 \newcommand*{\floatplacement}[2]{%
13 \SetupFloatingEnvironment{#1}{placement=#2}%
14 }
```

```
\floatstyle {<style>}
  Float styles are ignored.

15 \newcommand{\floatstyle}[1]{%
16 }

\restylefloat * {<style>}
  Float styles are ignored.

17 \NewDocumentCommand{\restylefloat}{s m}{%
18 }
```

File 49 **l warp-floatflts.sty**§ 125 Package **floatflts**

Pkg **floatflts** Emulated.
for HTML output: Discard all options for l warp-floatflts:

```
1 \LWR@ProvidesPackageDrop{floatflts}
```

Borrowed from the l warp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflts@marginfloat}{O{-1.2ex} m}
3 {%
4 \start
5 \LWR@maybeinthisfloat%
6 \LWR@forcenewpage%
7 \LWR@htmntag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
10 }
11 {
12 \LWR@htmldivclassend{div}
13 }
14
15 \DeclareDocumentEnvironment{floatingfigure}{o m}
16   {\begin{KFLTfloatflts@marginfloat}{figure}}
17   {\end{KFLTfloatflts@marginfloat}}
18
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20   {\begin{KFLTfloatflts@marginfloat}{table}#2}
21   {\end{KFLTfloatflts@marginfloat}}
```

File 50 **lwarf-floatpag.sty**

§ 126 Package **floatpag**

Pkg **floatpag** Emulated.

for HTML output: Discard all options for lwarf-floatpag:

```
1 \LWR@ProvidesPackageDrop{floatpag}
2 \newcommand*{\floatpagestyle}[1]{}
3 \newcommand*{\rotfloatpagestyle}[1]{}
4 \newcommand*{\thisfloatpagestyle}[1]{}
```

File 51 **lwarf-floatrow.sty**

§ 127 Package **floatrow**

Pkg **floatrow** floatrow is emulated during HTML output, and the floatrow package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{floatrow}
```

⚠ **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. lwarf 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:

```

2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }

\floobox [{<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`.

```

9 \NewDocumentCommand{\floobox}{o m o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}{%
11 {%

```

For `subcaption`:

```

12 \ifbool{LWR@insubfloatrow}{%
13 {%

```

`subfigure` and `subtable` environments take width as an argument.

```

14 \IfValueTF{#3}{%
15 {%
16 {%
17 }%
18 {%

```

`figure` and `table` environments do not take a width argument.

```

19 {%
20 }%
21 #6
22
23 #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}{%
25 {%
26 {%
27 }%
28 {%

```

For subfig:

```
29 \ifbool{LWR@insubfloatrow}{%
30 {%
31   \subfloat[#6]{#7}%
32 }%
33 {%
34   \begin{group}%
35     \let\caption\firstofone%
36     \subfloat[#6]{#7}%
37   \endgroup%
38 }%
39 {%
40   \subfloat[#6]{#7}%
41 }%
42 }%
43 {%
44 }
```

\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   \begin{group}%
32     \let\caption\firstofone%
33     \subfloat[#6]{#7}%
34   \endgroup%
35 }%
36 {%
37 }
```

figure and table are environments:

```
37 \Cnameuse{#2}%
38 #6%
39 %
40 #7%
41 \Cnameuse{end#2}%
42 }%
43 {%
44 }
```

Not used:

```
45 \newcommand*\nocapbeside{}%
46 \newcommand*\capbeside{}%
47 \newcommand*\caption{}%
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]{}%
56 \newcommand*\killfloatstyle{}%
```

Preamble and default width are ignored.

```
57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \Cnamedef{#1}{}}
```

```

59 \floatbox{#2}
60 }
61 }
```

Preamble and default width are ignored.

```

62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 @namedef{#1}{%
64 \floatbox{#2}{%
65 }%
66 }}

67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []
68 \newfloatcommand{ttabbox}{table}[\caption] [\FBwidth]
69 \newfloatcommand{fcapside}{figure}[\capbeside] []
```

The row of floats is placed into a <div> of class floatrow.

```

70 \newenvironment*{floatrow}[1][2]
71 {
72 \LWR@forcenewpage
73 \BlockClass{floatrow}
```

While inside the floatrow, divide the \linewidth by the number of floats.

```

74 \booltrue{\LWR@infloatrow}
75 \setlength{\linewidth}{6in/#1}
76 }
77 {
78 \boolfalse{\LWR@infloatrow}
79 \endBlockClass
80 }
```

Keys for \DeclareNewFloatType:

```

81 \newcommand*{\LWR@frowkeyplacement}{}
82 \newcommand*{\LWR@frowkeyname}{}
83 \newcommand*{\LWR@frowkeyfileext}{}
84 \newcommand*{\LWR@frowkeywithin}{}
85 \newcommand*{\LWR@frowkeycapstyle}{}
86
87 \define@key{frowkeys}{placement}{\relax}%
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{\#1}}%
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{\#1}}%
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{\#1}}%
91 \define@key{frowkeys}{relatedcapstyle}{\relax}%
```

Use `\listof{type}{Title}` to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
93 \renewcommand*{\LWR@frowkeyplacement}{}%
94 \renewcommand*{\LWR@frowkeyname}{}%
95 \renewcommand*{\LWR@frowkeyfileext}{}%
96 \renewcommand*{\LWR@frowkeywithin}{}%
97 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```
98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```
101 \ifthenelse{\equal{\LWR@frowkeywithin}{}}
102 {
103     \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement \
104                 \LWR@frowkeyfileext}%
105     \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}
106 }%
107 {%
108     \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement \
109                 \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110     \newfloat{#1}{\LWR@frowkeyplacement}%
111     {\LWR@frowkeyfileext}[\LWR@frowkeywithin]%
112     \LWR@traceinfo{finished newfloat #1}
113 }%
```

Rename the float if a name was given:

```
114 \ifthenelse{\equal{\LWR@frowkeyname}{}}
115 {}
116 {\floatname{#1}{\LWR@frowkeyname}}%
117 }
```

Not used:

```
118 \newcommand{\buildFBBOX}[2]{}
119 \newcommand*{\CenterFloatBoxes}{}%
120 \newcommand*{\TopFloatBoxes}{}%
121 \newcommand*{\BottomFloatBoxes}{}%
122 \newcommand*{\PlainFloatBoxes}{}%
123
```

```

124 \newcommand{\capsubrowsettings}{}
125
126 \NewDocumentCommand{\RawFloats}{o o}{}

```

To be used inside a minipage or parbox.

```
127 \newcommand{\RawCaption}[1]{#1}
```

Places additional text inside a float, inside a css <div> of class floatfoot.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{%
129 \begin{BlockClass}{floatfoot}
130 #2
131 \end{BlockClass}
132 }

```

Used to compute \linewidth.

```

133 \newbool{LWR@insubffloatrow}
134 \boolfalse{LWR@insubffloatrow}

```

```

135 \newenvironment*{subffloatrow}[1][2]
136 {

```

The row of floats is placed into a <div> of class floatrow:

```

137 \LWR@forcenewpage
138 \BlockClass{floatrow}

```

While inside the floatrow, LWR@insubffloatrow is set true, which tells \floatbox to use \subfigure or \subtable.

```

139 \begingroup
140 \booltrue{LWR@insubffloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@insubffloatrow}
146 }

```

File 52 lwarf-flushend.sty

§ 128 Package **flushend**

Pkg flushend Emulated.

for HTML output: Discard all options for lwarf-flushend:

```

1 \LWR@ProvidesPackageDrop{flushend}
2 %     \end{macrocode}
3 %
4 %     \begin{macrocode}
5 \newcommand*{\flushend}{}%
6 \newcommand*{\raggedend}{}%
7 \newcommand*{\flushcolsend}{}%
8 \newcommand*{\raggedcolsend}{}%
9 \newcommand*{\atColsBreak}[1]{}%
10 \newcommand*{\atColsEnd}[1]{}%
11 \newcommand*{\showcolsendlrule}{}%
```

File 53 **lwarf-fncychap.sty**

§ 129 Package **fncychap**

Pkg fncychap fncychap is ignored.

for HTML output: Discard all options for lwarf-fncychap:

```

1 \LWR@ProvidesPackageDrop{fncychap}

2 \def\mghrulefill#1{}
3 \def\ChNameLowerCase{}
4 \def\ChNameUpperCase{}
5 \def\ChNameAsIs{}
6 \def\ChTitleLowerCase{}
7 \def\ChTitleUpperCase{}
8 \def\ChTitleAsIs{}
9 \newcommand{\ChRuleWidth}[1]{}
10 \newcommand{\ChNameVar}[1]{}
11 \newcommand{\ChNumVar}[1]{}
12 \newcommand{\ChTitleVar}[1]{}
13 \newcommand{\TheAlphaChapter}{}
14 \newcommand{\DOCH}{}
15 \newcommand{\DOTI}[1]{}
16 \newcommand{\DOTIS}[1]{}
17 \newlength{\mylen}
18 \newlength{\myhi}
19 \newlength{\px}
20 \newlength{\py}
21 \newlength{\pyy}
22 \newlength{\pxx}
23 \newlength{\RW}
24 \newcommand{\FmN}[1]{#1}
25 \newcommand{\FmTi}[1]{#1}
```

File 54 l warp- fontenc .sty**§ 130 Package fontenc**

Pkg **fontenc** Error if fontenc is loaded after l warp.

Discard all options for l warp- fontenc:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontenc}

2 \LWR@loadbefore{fontenc}

File 55 l warp- fonts spec .sty**§ 131 Package fonts spec**

Pkg **fonts spec** Error if fonts spec is loaded after l warp.

Discard all options for l warp- fonts spec:

for HTML output: 1 \LWR@ProvidesPackageDrop{fonts spec}

2 \LWR@loadbefore{fonts spec}

File 56 l warp- footm istic .sty**§ 132 Package footm istic**

(Based on original code by ROBIN FAIRBAIRNS.)

Pkg **footm istic** footm istic is emulated during HTML output, and the footm istic package is ignored.

1 \LWR@ProvidesPackageDrop{footm istic}

Some nullified commands:

2 \newcommand{\footnotelayout}{}
3 \newcommand{\setfn symbol}[1]{}
4 \NewDocumentCommand{\DefineFNsymbols}{s m o m}{}

```

5
6 \newdimen\footnotemargin
7 \footnotemargin1.8em\relax
8
9 \newcommand*\hangfootparskip{0.5\baselineskip}
10 \newcommand*\hangfootparindent{0em}%
11
12 \let\pagefootnoterule\footnoterule
13 \let\mpfootnoterule\footnoterule
14 \def\splitfootnoterule{\kern-3\p0 \hrule \kern2.6\p0}
15
16 \providecommand*{\multiplefootnotemarker}{3sp}
17 \providecommand*{\multfootsep}{,}

```

Using cleveref:

```
18 \providecommand*{\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```

19 \newcommand\mpfootnotemark{%
20   \@ifnextchar[%%
21     \c@xmpfootnotemark
22   {%
23     \stepcounter\c@mpfn
24     \protected@edef\@thefnmark{\thempfn}%
25     \c@footnotemark
26   }%
27 }
28 \def\c@xmpfootnotemark[#1]{%
29   \begingroup
30   \csname c@\c@mpfn\endcsname #1\relax
31   \unrestored@protected@edef\@thefnmark{\thempfn}%
32   \endgroup
33   \c@footnotemark
34 }

```

File 57 **lwarf-footnote.sty**

§ 133 Package **footnote**

Pkg **footnote** footnote is used with minor patches.

for HTML output: 1 \LWR@ProvidesPackagePass{footnote}

Removed print-version formatting:

```

2 \def\fn@startnote{%
3 %   \parboxrestore%
4   \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
5 %   \color@begingroup% *** conflicts with lwarp
6 }
7
8 % \let\fn@endnote\color@endgroup% *** conflicts with lwarp
9 \newcommand*{\fn@endnote}{%
10 \LWR@htmltagc{/LWR@tagregularparagraph}%
11 \LWR@orignewline%
12 }

```

Removed print-version formatting:

```

13 \def\fn@startfntext{%
14   \setbox\z@\vbox\bgroup%
15     \fn@startnote%
16     \fn@prefntext%
17     \ignorespaces%
18 }

```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20   \LWR@htmltagc{/LWR@tagregularparagraph}%
21   \LWR@orignewline%
22   \fn@postfntext%
23   \egroup%
24   \begin{group}%
25     \let\@makefntext\empty%
26     \let\@finalstrut\gobble%
27     \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
28     \footnotetext{\unvbox\z@}%
29   \end{group}%
30 }

```

These have been redefined, so re-\let them again:

```

31 \let\endfootnote\fn@endfntext
32 \let\endfootnotetext\endfootnote

```

File 58 **lwarp-footnotehyper.sty**

§ 134 Package **footnotehyper**

Pkg **footnotehyper** **footnotehyper** is a hyperref-safe version of footnote. For **lwarp**, **footnotehyper** is

emulated.

for HTML output: Discard all options for l warp-footnotehyper:

```
1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}
```

File 59 **l warp-framed.sty**

§ 135 Package **framed**

(Based on original code by DONALD ARSENEAU.)

Pkg **framed** framed is supported and patched by l warp.

for HTML output: Accept all options for l warp-framed:

```
1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}%
3 % for \convertcolorspec
4 \renewenvironment{framed}{%
5 \LWR@forcenewpage
6 \BlockClass{framed}%
7 }
8 {\endBlockClass}
9
10 \renewenvironment{oframed}{%
11 \LWR@forcenewpage
12 \BlockClass{framed}%
13 }
14 {\endBlockClass}
15
16
17 \renewenvironment{shaded}{%
18 \convertcolorspec[named]{shadecolor}{HTML}\LWR@tempcolor%
19 \LWR@forcenewpage
20 \BlockClass[background: \#\LWR@tempcolor]{framed}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
25 \convertcolorspec[named]{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \#\LWR@tempcolor]{framed}%
28 }
29 {\endBlockClass}
30
```

```
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage
34     \BlockClass{framedleftbar}
35     \def\FrameCommand{}%
36     \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
45 }%
46 {\endBlockClass}
47
48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
52 }%
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{}{Opt}{Opt}{Opt}{Opt}{\BODY}
61 }

\CustomFBox {\langle toptitle\rangle} {\langle bottitle\rangle} {\langle thicknessstop\rangle} {\langle bottom\rangle} {\langle left\rangle} {\langle right\rangle} {\langle text contents\rangle}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{}{%
67     \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68         \textcolor{TFTitleColor}{\textbf{#1}}%
69     \end{BlockClass}%
70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{}{%
75     \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
```

```

76 \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77 \textcolor{TFTintColor}{\textbf{\#2}}}%
78 \end{BlockClass}%
79 }% not empty
80 \end{BlockClass}
81 }

\TitleBarFrame [marker] {title} {contents}

82 \renewcommand\TitleBarFrame[3][]{%
83 \CustomFBox
84 {\#2}{}}%
85 \fboxrule\fboxrule\fboxrule\fboxrule
86 {\#3}%
87 }

88 \renewcommand{\TF@Title}[1]{#1}

MakeFramed {settings}

89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\ linewidth}\BODY\end{minipage}}%
94 }

\fb@put@frame {frame cmd no split} {frame cmd split}

95 \renewcommand*{\fb@put@frame}[2]{%
96 \relax%
97 \tempboxa%
98 }

```

File 60 **lwarp-ftnright.sty**

§ 136 Package **ftnright**

Pkg **ftnright** ftnright is ignored.

for HTML output: Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

File 61 **l warp-fullpage.sty**

§ 137 Package **fullpage**

Pkg **fullpage** Emulated.

for HTML output: Discard all options for l warp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

File 62 **l warp-fullwidth.sty**

§ 138 Package **fullwidth**

Pkg **fullwidth** **fullwidth** is emulated.

A minipage is used, of no HTML width.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fullwidth}

2 \newenvironment*{\fullwidth}[1] []{%
3 \minipagewidth%
4 \minipage{\ linewidth}%
5 }
6 {%
7 \endminipage%
8 }
```

File 63 **l warp-geometry.sty**

§ 139 Package **geometry**

Pkg **geometry** **geometry** is preloaded by l warp, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for l warp-geometry:

```
1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*{\geometry}[1]{}
```

```

3 \renewcommand*{\newgeometry}[1]{}
4 \renewcommand*{\restoregeometry}(){}
5 \renewcommand*{\savegeometry}[1]{}
6 \renewcommand*{\loadgeometry}[1]{}

```

File 64 **lwarf-glossaries.sty**

§ 140 Package **glossaries**

Pkg **glossaries** xindy is required for glossaries.

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

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

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 HTMLpage:

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

```

Opt **IndexLanguage** The lwarf package takes an option `IndexLanguage=english` to set the language used by xindy. This is passed to xindy using its `-L` option, and is used for both index and glossary generation.

Opt **lwarfmk printglossary** lwarfmk has the commands `lwarfmk printglossary` and `lwarfmk htmlglossary` to process the glossaries created by glossaries using xindy.

for HTML output: 1 `\PassOptionsToPackage{xindy}{glossaries}`

```

2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```

5 \renewcommand*{\@p@glossarysection}[2]{%
6   \glsclearpage
7   \phantomsection
8   \ifdef\empty\@glossarysecstar
9   {%
10     \csname\@glossarysec\endcsname{#2}%
11   }%
12 }
```

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. Here, the TOC entry is made after the section is created:

```

13   \csname\@glossarysec\endcsname*{#2}%
14   \gls@toc{#1}{\@glossarysec}% Moved after the previous line.
15 }%
16 \glossaryseclabel
17 }
```

File 65 lwarf-graphics.sty

§ 141 Package **graphics**

Pkg **graphics** **graphics** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphics}

File 66 lwarf-graphicx.sty

§ 142 Package **graphicx**

Pkg **graphicx** **graphicx** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{graphicx}

File 67 **lwarf-grffile.sty**

§ 143 Package **grffile**

Pkg grffile grffile is partially emulated. 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 version for HTML. No attempt is made to convert input or filename encodings.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{grffile}

2 \define@key{igraph}{multidot}={}
3 \define@key{igraph}{babel}={}
4 \define@key{igraph}{extendedchars}={}
5 \define@key{igraph}{encoding}={}
6 \define@key{igraph}{inputencoding}={}
7 \define@key{igraph}{filenameencoding}={}
8 \define@key{igraph}{space}={}
9
10 \newcommand*\grffilessetup[1]{}

```

File 68 **lwarf-hyperref.sty**

§ 144 Package **hyperref**

Pkg hyperref hyperref is emulated during HTML output, and the hyperref package is ignored.

for HTML output:

```

1 \% \LWR@ProvidesPackageDrop{hyperref}
2 \typeout{Using the lwarf html version of package 'hyperref' -- discarding options.}
3 \typeout{ Are not using ProvidesPackage, so that other packages}
4 \typeout{ do not attempt to patch lwarf's version of 'hyperref'.}
5 \% \ProvidesPackage{lwarf-\#1-\#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*\hypersetup[1]{}
9 \newcommand*\hyperbaseurl[1]{}

```

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@htmntag{img src="#1" alt="#2" class="hyperimage"}%}

```

```
12 \endgroup%
13 \LWR@ensuredoingapar%
14 }
15
16 \newcommand{\hyperimage}{%
17 \LWR@ensuredoingapar%
18 \begingroup\catcode`\_=12
19 \LWR@hyperimageb%
20 }
```

Creates an HTML anchor to category.name with the given text.

```
21 \NewDocumentCommand{\hyperdef}{m m +m}{%
22 \LWR@ensuredoingapar%
23 \LWR@subsublabel{#1.#2}%
24 #3%
25 }
```

Creates an HTML link to URL#category.name with the given text.

```
26 \NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27 \LWR@htmltag{a href="#1\LWR@hashmark#2.#3"}%
28 #4%
29 \LWR@htmltag{/a}%
30 \endgroup%
31 }
```

Creates text as an HTML link to the L^AT_EX label.

```
32 \NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{%
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }

38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode`\_=12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }
```

Creates an anchor to name with the given text.

```
43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }
```

Creates a link to the anchor created by hypertarget, with the given link text.

```
47 \NewDocumentCommand{\hyperlink}{m +m}{%
48   \hyperref [#1]{#2}%
49 }
```

For HTML, \cleveref is used instead.

```
50 \NewDocumentCommand{\autoref}{s m}{%
51   \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }
```

For HTML, \cleveref is used instead.

```
53 \NewDocumentCommand{\autopageref}{s m}{%
54   \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }

56 \newcommand{\pdfstringdef}[2] {}

57 \newcommand{\pdfbookmark}[3] [] {}

58 \newcommand{\currentpdfbookmark}[2] {}

59 \newcommand{\subpdfbookmark}[2] {}

60 \newcommand{\belowpdfbookmark}[2] {}

61 \newcommand{\texorpdfstring}[2]{#2}
```

From hyperref.

```
62 \def\hypercalcbp#1{%
63   \strip@pt\dimexpr 0.99626401\dimexpr (#1)\relax\relax
64 }%

65 \newcommand{\Acrobatmenu}[2] {}

66 \newcommand*\{\TextField}[2] [] {}

67 \newcommand*\{\CheckBox}[2] [] {}

68 \newcommand{\ChoiceMenu}[3] [] {}

69 \newcommand*\{\PushButton}[2] [] {}}
```

```
70 \newcommand*{\Submit}[2] {}  
71 \newcommand*{\Reset}[2] {}  
72 \newcommand*{\LayoutTextField}[2] {}  
73 \newcommand*{\LayoutChoiceField}[2] {}  
74 \newcommand*{\LayoutCheckField}[2] {}  
75 \newcommand*{\MakeRadioField}[2] {}  
76 \newcommand*{\MakeCheckField}[2] {}  
77 \newcommand*{\MakeTextField}[2] {}  
78 \newcommand*{\MakeChoiceField}[2] {}  
79 \newcommand{\MakeFieldButton}[1] {}
```

File 69 **lwarf-hyperxmp.sty**

§ 145 Package **hyperxmp**

Pkg **hyperxmp** Emulated.

for HTML output: Discard all options for lwarf-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 70 **lwarf-idxlayout.sty**

§ 146 Package **idxlayout**

Pkg **idxlayout** Emulated.

for HTML output: Discard all options for lwarf-idxlayout:

```
1 \LWR@ProvidesPackageDrop{idxlayout}  
2 \newcommand{\LWR@indexprenote}{}  
3
```

```
4 \renewcommand*\{\printindex}
5 {
6 \LWR@startpars
7
8 \LWR@indexprenote
9
10 \LWR@origprintindex
11 }
12
13 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
14 \newcommand*{\noindexprenote}{\renewcommand{\LWR@indexprenote}{\relax}}
15
16 \newcommand{\idxlayout}[1]{}
17 \newcommand*{\indexfont}{}
18 \newcommand*{\indexjustific}{}
19 \newcommand*{\indexsubsdelim}{}
20 \newcommand*{\indexstheadcase}{}
```

File 71 **lwarp-indentfirst.sty**

§ 147 Package **indentfirst**

Pkg **indentfirst** **indentfirst** is ignored.

Discard all options for **lwarp-indentfirst**:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

File 72 **lwarp-inputenc.sty**

§ 148 Package **inputenc**

Pkg **inputenc** Error if **inputenc** is loaded after **lwarp**.

Discard all options for **lwarp-inputenc**:

for HTML output: 1 \LWR@ProvidesPackageDrop{inputenc}
 2 \LWR@loadbefore{inputenc}

File 73 l warp-keyfloat.sty

§ 149 Package **keyfloat**

Pkg keyfloat keyfloat is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{keyfloat}

After keyfloat has loaded:

```
2 \AtBeginDocument{

3 \let\KFLT@boxinner\relax
4 \let\endKFLT@boxinner\relax
5
6 \NewEnviron{KFLT@boxinner}
7 {%
8 \LWR@traceinfo{kflt@boxinner}%
9 \LWR@stoppars%
10 \KFLT@frame{\BODY}%
11 \LWR@startpars%
12 \LWR@traceinfo{ended kflt@boxinner}%
13 }

14 \DeclareDocumentEnvironment{KFLT@marginfloat}{O{-1.2ex} m}
15 {%
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmlltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 \LWR@htmldivclassend{div}
25 }

26 \DeclareDocumentEnvironment{marginfigure}{o}
27 {\begin{KFLT@marginfloat}{figure}}
28 {\end{KFLT@marginfloat}}
29
30 \DeclareDocumentEnvironment{margintable}{o}
31 {\begin{KFLT@marginfloat}{table}}
32 {\end{KFLT@marginfloat}}
```

```
33 \DeclareDocumentEnvironment{keywrap}{m +m}
34 {%
35 \begin{BlockClass}{marginblock}
36 \setlength{\ linewidth}{#1}
37 #2%
38 \end{BlockClass}
39 }
40 {%
41 }
```

42 }% AtBeginDocument

File 74 l warp-layout.sty**§ 150 Package layout**

Pkg layout layout is ignored.

for HTML output: Discard all options for l warp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}{}
```

File 75 l warp-letterspace.sty**§ 151 Package letterspace**

Pkg letterspace letterspace is a subset of microtype, which is pre-loaded by l warp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for l warp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2][]{}
4 \def\textls#1#1{#1}
5 \newcommand*\lslig[1]{#1}
```

File 76 **l warp-lettrine.sty**

§ 152 Package **lettrine**

(Based on original code by DANIEL FLIPO.)

Pkg **lettrine** Emulated.

for HTML output: Discard all options for l warp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a of class **lettrine**, and the following text is in a of class **lettrinetext**. \lettrine [<keys>] {<letter>} {<additional text>}

```
2 \DeclareDocumentCommand{\lettrine}{o m m}{%
3 \InlineClass{lettrine}{#2}\InlineClass{lettrinetext}{#3} %
4 }
5
6 \newcounter{DefaultLines}
7 \setcounter{DefaultLines}{2}
8 \newcounter{DefaultDepth}
9 \newcommand*\{\\DefaultOptionsFile\}{\relax}
10 \newcommand*\{\\DefaultLoversize\}{0}
11 \newcommand*\{\\DefaultLraise\}{0}
12 \newcommand*\{\\DefaultLhang\}{0}
13 \newdimen\DefaultFindent
14 \setlength{\DefaultFindent}{\z@}
15 \newdimen\DefaultNindent
16 \setlength{\DefaultNindent}{0.5em}
17 \newdimen\DefaultSlope
18 \setlength{\DefaultSlope}{\z@}
19 \newdimen\DiscardVskip
20 \setlength{\DiscardVskip}{0.2\p@}
21 \newif\ifLettrineImage
22 \newif\ifLettrineOnGrid
23 \newif\ifLettrineRealHeight
24
25 \newcommand*\{\\LettrineTextFont\}{\scshape}
26 \newcommand*\{\\LettrineFontHook\}{}
27 \newcommand*\{\\LettrineFont\}[1]{\InlineClass{lettrine}{#1}}
28 \newcommand*\{\\LettrineFontEPS\}[1]{\includegraphics[height=1.5ex]{#1}}
```

File 77 **l warp-lips.sty**

§ 153 Package **lips**

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```

1 \% \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{l warp}{Using the l warp version of package 'lips'.}%
3 \ProvidesPackage{l warp-lips}
4
5 \NewDocumentCommand{\Lips}{}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}
8
9 \let\lips\Lips
10 \let\oldlips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{%
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}

```

File 78 **l warp-listings.sty**

§ 154 Package **listings**

(Based on original 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: 1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{listings}

Patches to embed listings inside pre tags:

```

3 \let\LWR@origlst@Init\lst@Init
4 \let\LWR@origlst@DeInit\lst@DeInit
5

```

```

6 \let\LWR@origlsthkEveryPar\lsthk@EveryPar
7
8 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{lstlisting}{\l@lstlisting}{#1}{#2}}

```

Done at the start of a listing.

```
9 \renewcommand{\lst@Init}[1]{%
```

First, perform the listings initialization:

```

10 \LWR@traceinfo{lst@Init}%
11 \renewcommand*{\@capttype}{lstlisting}%
12 \LWR@origlst@Init{#1}%
13 \LWR@traceinfo{finished origlst@Init}%
14 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```

15 \LWR@traceinfo{About to create verbatim.}%
16 \let\lsthk@EveryPar\relax%
17 \LWR@forcenewpage
18 \LWR@atbeginverbatim{programlisting}%
19
20 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
21 \else%

```

Inline, so open a :

```

22 \ifbool{\LWR@verbtags}{\LWR@htmntag{span class="inlineprogramlisting"}{}}{%
23 \fi%
24 }

25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```

27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%

```

Inline, so create the closing :

```
32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%
```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarf.

```
36 \def\lst@MakeCaption#1{%
37 \LWR@traceinfo{MAKING CAPTION at #1}%
38   \lst@ifdisplaystyle
39 \LWR@traceinfo{making a listings display caption}%
40   \ifx #1t%
41     \ifx\lst@caption\@empty\expandafter\lst@HRefStepCounter \else
42       \expandafter\refstepcounter
43     \fi {\lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 %
46 % \label{\lst@label}\fi
47 \LWR@traceinfo{Finished assigning the label.}%
48   \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
49   \global\let\lst@name\lst@arg \global\let\lstname\lst@name
50   \lst@ifnolol\else
51     \ifx\lst@@caption\@empty
52       \ifx\lst@caption\@empty
53         \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54           \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarf:

```
55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %
57   \addcontentsline{loll}{\lstlisting}{\lst@name}
58   \fi\fi
59   \else
```

This would have to be modified for lwarf:

```
60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61   \addcontentsline{loll}{\lstlisting}{%
62 {\protect\numberline{\thelstlisting}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64   \fi
65   \fi
```

```

66      \fi
67      \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{\lst@caption not empty-}%
69      \lst@ifSubstring #1\lst@captionpos
70          {\begingroup
71 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

72 \%           \let\@vskip\vskip
73 \%           \def\vskip{\afterassignment\lst@vskip \tempskipa}%
74 \%           \def\lst@vskip{\nobreak\@vskip\tempskipa\nobreak}%
75 \%           \par\@parboxrestore\normalsize\normalfont \% \noindent (AS)
76 \%           \ifx #1\allowbreak \fi
77 \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

78           \lst@makecaption\fnum@lstlisting{\ignorespaces \lst@caption}
79       \else

```

New lwarp code to create a title:

```

80 \%           \lst@maketitle\lst@title \% (AS)
81 \LWR@traceinfo{Making title: \lst@title}%
82 \begin{BlockClass}{\lstlistingtitle}\% lwarp
83 \lst@maketitle\lst@title\% lwarp
84 \end{BlockClass}\% lwarp
85      \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87      \ifx\lst@label\@empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef@\currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for lwarp:

```

94 \%           \ifx #1\allowbreak \fi
95           \endgroup\}{}\%
96      \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98      \else
99 \LWR@traceinfo{INLINE}%
100     \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

103 \lst@Key{numbers}{none}%
104     \let\lst@PlaceNumber\empty
105     \lstKV@SwitchCases{#1}%
106     {none&\%
107         left&\def\lst@PlaceNumber{%
108 % \llap{%
109 \LWR@orignormalfont%
110 \lst@numberstyle{\the\lstnumber}\kern\lst@numbersep%
111 }%
112 }%
113 \\%
114     right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont%
115             \kern\VerbatimHTMLWidth \kern\lst@numbersep%
116             \lst@numberstyle{\the\lstnumber}}}}%
117 }{\PackageError{Listings}{Numbers #1 unknown}@ehc}%
118 \end{warpHTML}

```

File 79 lwarp-longtable.sty

§ 155 Package **longtable**

Pkg **longtable** longtable is emulated during HTML output, and the longtable package is ignored.

for HTML output: 1 \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 relevant to HTML output.

 \kill is ignored, place a \kill line inside

\begin{warpprint} ... \end{warpprint}

or place it inside \warpingprintonly.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` * [*horizontalignment*] {*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.

```

2 \newenvironment{longtable*}[2] []{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{\LTcaptype}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{\LWR@starredlongtable}%
9 \let\captionlistentry{\LWR@LTcaptionlistentry}%
10 \LWR@tabular{#2}%
11 }%
12 {\endLWR@tabular\endLWR@floatend}%
13
14 \newenvironment{longtable}[2] []{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaptype}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaptype}%
21 \let\captionlistentry{\LWR@LTcaptionlistentry}%
22 \LWR@tabular{#2}%
23 }%
24 {\endLWR@tabular\endLWR@floatend}%
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendofline}% throws away options // [dim] and //*
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTright}
35 \newlength{\LTpre}

```

```
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}
38 \renewcommand*{\kill}{\LWR@tabularendofline}
```

File 80 **lwarf-lscape.sty**

§ 156 Package **lscape**

Pkg **lscape** **lscape** is nullified.

for HTML output: Discard all options for **lwarf-lscape**.

```
1 \LWR@ProvidesPackageDrop{lscape}
2 \newenvironment*{landscape}{}{}
```

File 81 **lwarf-ltcaption.sty**

§ 157 Package **ltcaption**

Pkg **ltcaption** **ltcaption** is emulated during HTML output, and the **ltcaption** package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcaptype is already defined by **lwarf**.

longtable* is already defined by **lwarf-longtable**.

```
2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginsfalse}{}{}
```

File 82 **lwarf-marginfit.sty**

§ 158 Package **marginfit**

Pkg **marginfit** Emulated.

for HTML output:

Discard all options for l warp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}
```

File 83 **l warp-marginfix.sty**

§ 159 Package **marginfix**

Pkg **marginfix** Emulated.

for HTML output: Discard all options for l warp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][]{}
10 \newcommand*{\unblockmargin}[1][]{}
11 \newcommand*{\marginphantom}[2][]{}
```

File 84 **l warp-marginnote.sty**

§ 160 Package **marginnote**

Pkg **marginnote** Emulated.

for HTML output: Discard all options for l warp-marginnote:

```
1 \LWR@ProvidesPackageDrop{marginnote}

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{\#2}}
3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}
8 \newcommand*{\marginfont}{}
9 \newcommand*{\raggedleftmarginnote}{}
10 \newcommand*{\raggedrightmarginnote}{}
```

File 85 **lwarp-mcaption.sty**

§ 161 Package **mcaption**

Pkg **mcaption** **mcaption** is nullified.

for HTML output: Discard all options for **lwarp-mcaption**:

```
1 \LWR@ProvidesPackageDrop{mcaption}
2 \newenvironment{margincap}{}{}
3 \newcommand*\margincapalign{}
4 \newlength{\margincapsep}
```

File 86 **lwarp-mdframed.sty**

§ 162 Package **mdframed**

Pkg **mdframed** **mdframed** is loaded with options forced to **framemethod=none**.

for HTML output:

```
1 \RequirePackage{xcolor}%
2 \LWR@ProvidesPackageDrop{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.

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 `` of class `mdframedtitle`. Subtitles are in a `` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

`amsthm` must be loaded before `mdframed`

```
3 \LWR@origRequirePackage{amsthm}
```

Do not require `Tikz` or `pstricks`:

```
4 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

To handle CSS and paragraphs, patch code at start and end of environment and contents. `\LWR@origraggedright` helps avoid hyphenation.

```
5 \mdfsetup{
6 startcode={\LWR@mdframedstart\LWR@origraggedright},
7 endcode={\LWR@mdframedend},
8 startinnercode={\LWR@startpars\LWR@origraggedright},
9 endinnercode={\LWR@stopars},
10 }
```

Given the `mdframed` key, print the color.

```
11 \newcommand*{\LWR@mdfprintcolor}[1]{%
12 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
13 \#\LWR@tempcolor
14 }
```

Given the `mdframed` key, print the length.

```
15 \newcommand*{\LWR@mdfprintlength}[1]{%
16 \rndprintlength{\csuse{mdf@#1@length}}
17 }
```

Actions before an `mdframe` starts.

Encapsulate a frame inside a <div> of the desired class.

```
18 \newcommand*{\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
19 \LWR@stoppars%
```

Below, print HTML pt units:

```
20 \uselengthunit{PT}%
```

Open a <div> and with custom class and custom style:

```
21 \LWR@htmtagc{div class="\LWR@mdthisenv" \LWR@orignewline
22 style=" \LWR@orignewline
```

Convert and print the background color:

```
23 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
24 border: \LWR@mdfprintlength{linewidth} solid
25 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
26 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
27 \ifbool{mdf@shadow}{%
28   box-shadow:
29   \LWR@mdfprintlength{shadowsize}
30   \LWR@mdfprintlength{shadowsize}
31   \LWR@mdfprintlength{shadowsize}
32   \LWR@mdfprintcolor{shadowcolor} ;
33 }
34 {box-shadow: none ;}
35 \LWR@orignewline
36 "}
37 % \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`:

```
38 \LetLtxMacro{\hspace}{\LWR@orighspace}%
39 \LetLtxMacro{\rule}{\LWR@origrule}%
40 \LetLtxMacro{\makebox}{\LWR@origmakebox}%
41 }
```

Actions after an mdframe ends.

After closing the <div>, globally restore to the default environment type:

```
42 \newcommand*{\LWR@mdframedend}{
```

Close the custom <div>:

```
43 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
44 \gdef{\LWR@mdthisenv}{mdframed}
```

Resume paragraph handling:

```
45 \LWR@startpars%
46 }
```

Encapsulation of the original which places the title inside a of class mdframedtitle:

```
47 \LetLtxMacro{\LWR@origmdfframedtitleenv}{\mdfframedtitleenv}
48
49 \newlength{\LWR@titleroundcorner}
50
51 \renewrobustcmd{\mdfframedtitleenv}[1]{%
52 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
53 \uselengthunit{PT}%
```

Open a with a custom class and custom style:

```
54 \LWR@htmleagc{span class="mdframedtitle" \LWR@orignewline
55 style=" \LWR@orignewline
```

Convert and print the title background color:

```
56 background:
57 \LWR@mdfprintcolor{frametitlebackgroundcolor}
58 ; \LWR@orignewline
```

Convert and print the title rule:

```
59 \ifbool{mdf@frametitlerule}{%
60   border-bottom:
61   \LWR@mdfprintlength{frametitlerulewidth}
62   solid
63   \LWR@mdfprintcolor{frametitlerulecolor}
64   ; \LWR@orignewline
65 }{}}
```

The title's top border radius is adjusted for the line width:

```
66 border-radius:
67 \setlength{\LWR@titleroundcorner}
68   {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{0pt}}
69   \rndprintlength{\LWR@titleroundcorner}
70   \rndprintlength{\LWR@titleroundcorner}
71   Opt Opt
72   \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
73 " \LWR@orignewline
74 }% span
```

Restrict paragraph tags inside a span:

```
75 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
76 #1%
```

Closee the span and unnest the paragraph tag restriction:

```
77 \LWR@htmlltagc{/span}%
78 \end{LWR@nestspan}%
79 }
80 }
```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `` of class `mdframedsubtitle`:

```
81 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
82 {%
83   the following empty line is required
84 }
```

Special handling for mdframed: Subtitles have \pars around them, so temporarily disable them here.

```
84 \let\par\LWR@origpar%
```

Open a with a custom class and custom style:

```
85 \LWR@htmltagc{span class="mdframed#1title"
86 style="" \LWR@orignewline
```

Convert and print the background color:

```
87 background:
88 \LWR@mdfprintcolor{#1titlebackgroundcolor}
89 ; \LWR@orignewline
```

Convert and print the above line:

```
90 \ifbool{mdf@#1titleaboveline}{%
91   border-top:
92   \LWR@mdfprintlength{#1titleabovelinewidth}
93   solid
94   \LWR@mdfprintcolor{#1titleabovelinecolor}
95   ; \LWR@orignewline
96 }{}}
```

Convert and print the below line:

```
97 \ifbool{mdf@#1titlebelowline}{%
98   border-bottom:
99   \LWR@mdfprintlength{#1titlebelowlinewidth}
100  solid
101  \LWR@mdfprintcolor{#1titlebelowlinecolor}
102  ; \LWR@orignewline
103 }{}}
```

Finish the custom style and the opening span tag:

```
104 "}% span
```

Restrict paragraph tags inside a span:

```
105 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```
106 \IfNoValueTF{#2}
107 {\csuse{LWR@origmdf#1title}{#3}}%
108 {\csuse{LWR@origmdf#1title}{[#2]{#3}}}%
```

Close the span and unnest the paragraph tag restriction:

```

109 \LWR@htmtagc{/span}%
110 \end{LWR@nestspan}%
111 } must follow the /span or an extra <p> appears

113 \let\LWR@origmdfsubtitle\mdfsubtitle
114
115 \newcommand*\{\LWR@mdfsubtitle}{%
116 \LWR@mdfsubtitlecommon{sub}%
117 }
118 \let\mdfsubtitle\LWR@mdfsubtitle

119 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
120
121 \newcommand*\{\LWR@mdfsubsubtitle}{%
122 \LWR@mdfsubtitlecommon{subsub}%
123 }
124 \let\mdfsubsubtitle\LWR@mdfsubsubtitle

```

Stores the environment of the frame about to be created:

```
125 \newcommand*\{\LWR@mdthisenv}{mdframed}
```

Modified from the original to remember the environment.

```

126 \renewrobustcmd*\newmdenv[2] [] {%
127 \newenvironment{#2}%
128 {%
129 \mdfsetup{#1}%
130 \renewcommand*\{\LWR@mdthisenv}{md#2}%
131 \begin{mdframed}%
132 }%
133 {\end{mdframed}}%
134 }
```

Modified from the original to remember the environment.

```

135 \renewrobustcmd*\{\surroundwithmdframed\}[2] [] {%
136 \BeforeBeginEnvironment{#2}%
137 \renewcommand*\{\LWR@mdthisenv}{md#2}%
138 \begin{mdframed}[#1]}%
139 \AfterEndEnvironment{#2}{\end{mdframed}}%
140 }
```

[*numberedlike*] {[*caption*} [*within*]}

Modified from the original to remember the environment.

```

141 \DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }%
142   {\@ifcsdef{#2}%
143    { \mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
144    {%
145      \IfNoValueTF {#3}%
146        { \IfNoValueTF {#5}%
147          { \IfNoValueTF {#5}%
148            { \IfNoValueTF {#5}%
149              { \@definecounter{#2}%
150                \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
151                \newenvironment{#2}[1] []{%
152                  \refstepcounter{#2}%
153                  \ifstrempty{##1}%
154                    { \let\@temptitle\relax}%
155                    {%
156                      \def\@temptitle{\mdf@theoremseparator}%
157                      \mdf@theoremspace%
158                      \mdf@theoremtitlefont%
159                      ##1}%
160                      \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname}{##1}}%
161                    }%
162                  \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname}%
163                                \@temptitle}]%%
164                \end{mdframed}%
165                \newenvironment{#2*}[1] []{%
166                  \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
167                  \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]%%
168                  \end{mdframed}%
169                }%
170                { \#5 given -- reset counter%
171                  \@definecounter{#2}\@newctr{#2}{#5}%
172                  \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
173                  \expandafter\xdef\csname the#2\endcsname{%
174                    \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
175                    \@thmcounter{#2}}%
176                  \newenvironment{#2}[1] []{%
177                    \refstepcounter{#2}%
178                    \ifstrempty{##1}%
179                      { \let\@temptitle\relax}%
180                      {%
181                        \def\@temptitle{\mdf@theoremseparator}%
182                        \mdf@theoremspace%
183                        \mdf@theoremtitlefont%
184                        ##1}%
185                        \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname}{##1}}%
186                      }%
187                  \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname}%
188                                \@temptitle}]%%

```

```

189      {\end{mdframed}}%
190      \newenvironment{#2*}[1][]{%
191          \ifstrempty{##1}%
192              {\let\@temptitle\relax}%
193              {%
194                  \def\@temptitle{\mdf@theoremseparator%
195                                  \mdf@theoremspace%
196                                  \mdf@theoremtitlefont%
197                                  ##1}%
198                  \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname}{##1}}%
199              }%
200          \begin{mdframed}[#1,frametitle={\strut\#4\@temptitle}]%
201              {\end{mdframed}}%
202          }%
203      }%
204      {##3 given -- number relationship
205          \global\@namedef{the#2}{\nameuse{the#3}}%
206          \newenvironment{#2}[1][]{%
207              \refstepcounter{##3}%
208              \ifstrempty{##1}%
209                  {\let\@temptitle\relax}%
210                  {%
211                      \def\@temptitle{\mdf@theoremseparator%
212                                  \mdf@theoremspace%
213                                  \mdf@theoremtitlefont%
214                                  ##1}%
215                      \mdf@thm@caption{#2}{\#4}{\csname the#2\endcsname}{##1}}%
216                  }%
217          \begin{mdframed}[#1,frametitle={\strut\#4\ \csname the#2\endcsname%
218                                      \@temptitle}]%
219              {\end{mdframed}}%
220          \newenvironment{#2*}[1][]{%
221              \ifstrempty{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
222              \begin{mdframed}[#1,frametitle={\strut\#4\@temptitle}]%
223              \end{mdframed}}%
224          }%
225          \BeforeBeginEnvironment{#2}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
226          \BeforeBeginEnvironment{#2*}{\renewcommand*{\LWR@mdthisenv}{md#2}}% new
227      }%
228  }

```

[*numberedlike*] {[*caption*} [*within*]}

Modified from the original to remember the environment.

```

229 \DeclareDocumentCommand\newmdtheoremenv{O{} m o m o }{%
230     \ifboolexpr{ test {\IfNoValueTF{#3}} and test {\IfNoValueTF{#5}} }{%
231         {\newtheorem{#2}{#4}}{%
232             \IfValueTF{#3}{\newtheorem{#2}{#3}{#4}}{}%
233             \IfValueTF{#5}{\newtheorem{#2}{#4}{#5}}{}%

```

```

234      }%
235 \BeforeBeginEnvironment{#2}{%
236 \renewcommand*{\LWR@mdthisenv}{\md#2}%
237 \begin{mdframed}[#1]}%
238 \AfterEndEnvironment{#2}{%
239 \end{mdframed}}%
240 }

```

File 87 **lwarp-metalogo.sty**

§ 163 Package **metalogo**

Pkg **metalogo** metalogo is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{metalogo}

2 \newcommand\setlogokern[2]{}
3 \newcommand\setlogodrop[2][XeTeX]{}
4 \newcommand\setLaTeXa[1]{}
5 \newcommand\setLaTeXee[1]{}
6 \newcommand\seteverylogo[1]{}
7 \newcommand\everylogo[1]{}

```

File 88 **lwarp-microtype.sty**

§ 164 Package **microtype**

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:

```

1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}={}
3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m}={}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}={}
5 \DeclareDocumentCommand{\SetProtrusion}{o m m}={}
6 \DeclareDocumentCommand{\SetExpansion}{o m m}={}
7 \DeclareDocumentCommand{\SetTracking}{o m m}={}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m}={}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m}={}
10 \DeclareDocumentCommand{\DisableLigatures}{o m}={}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}={}

```

```

12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m} {}
13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m} {}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m} {}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m} {}
16 \DeclareDocumentCommand{\microtypesetup}{m} {}
17 \DeclareDocumentCommand{\microtypecontext}{m} {}
18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{o}{}
21 \DeclareDocumentCommand{\textls}{o +m}{}
22 \DeclareDocumentCommand{\lslig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1{\gobbletwo}
25 \def\DeclareMicrotypeVariants#1{\gobble}
26 \onlypreamble\DeclareMicrotypeSet
27 \onlypreamble\UseMicrotypeSet
28 \onlypreamble\DeclareMicrotypeSetDefault
29 \onlypreamble\DisableLigatures
30 \onlypreamble\DeclareMicrotypeVariants
31 \onlypreamble\DeclareMicrotypeBabelHook

```

File 89 **lwarf-moreverb.sty**

§ 165 Package **moreverb**

Pkg **moreverb** moreverb is supported with some patches.

for HTML output:

```

1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{moreverb}

3 \BeforeBeginEnvironment{verbatimtab}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{verbatimtab}{%
8 \LWR@afterendverbatim%
9 }
10
11
12 \LetLtxMacro{\LWRMV@orig@verbatimtabinput}{\@verbatimtabinput}
13
14 \renewcommand{\@verbatimtabinput}[2][]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatimtabinput[#1]{#2}%
18 \LWR@afterendverbatim%

```

```
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
23 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 % \unskip\LWR@origvspace*{-\baselineskip}%
28 \LWR@afterendverbatim%
29 }
30
31 \BeforeBeginEnvironment{listingcont}{%
32 \LWR@forcenewpage
33 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
34 }
35
36 \AfterEndEnvironment{listingcont}{%
37 % \unskip\LWR@origvspace*{-\baselineskip}%
38 \LWR@afterendverbatim%
39 }
40
41 \renewcommand{\listinglabel}[1]{\InlineClass{listinglabel}{\the#1}}
42
43 \def\thelisting@line{%
44 \@tempcnta=\listing@line%
45 \divide\@tempcnta\listing@step \multiply\@tempcnta\listing@step%
46 \ifnum\listing@line=\@ne%
47 \listinglabel\listing@line%
48 \else%
49 \ifnum\@tempcnta=\listing@line%
50 \listinglabel\listing@line%
51 \else%
52 \InlineClass{listinglabel}{}%
53 \fi%
54 \fi}
55
56 \LetLtxMacro{\LWRMV@}{\listinginput\@listinginput}
57
58 \renewcommand{\@listinginput}[3][]{%
59 \LWR@forcenewpage
60 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
61 \LWRMV@{\listinginput[#1]{#2}{#3}}
62 \LWR@afterendverbatim%
63 }
64
65
66 \renewenvironment*{boxedverbatim}{%
67 {
68 \LWR@forcenewpage
```

```
69 \LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-\baselineskip}%
70 \verbatim%
71 }%
72 {%
73 \endverbatim%
74 \LWR@afterendverbatim%
75 }%
76
77

78 \end{warpHTML}
```

File 90 **l warp-mparhack.sty**§ 166 Package **mparhack**

Pkg **mparhack** Emulated.

for HTML output: Discard all options for l warp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

File 91 **l warp-multicol.sty**§ 167 Package **multicol**

Pkg **multicol** multicol is emulated during HTML output, and the multicol package is ignored.

for HTML output: 1 \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`.

```
2 \begin{warpHTML}

3 \NewDocumentEnvironment{multicols}{s m o}
```

HTML <div> class to contain everything:

```
4 {
```

```

5 \LWR@forcenewpage
6 \BlockClass{multicols}

```

Optional HTML <div> class for the heading:

```
7 \IfValueTF{#3}{\begin{BlockClass}{multicolsheading}#3\end{BlockClass}}{}}
```

When done with the environment, close the <div>:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```

9 \newcommand*{\columnbreak}={}
10 \newcommand*{\Rmulticolcolumns}={}
11 \newcommand*{\Lmulticolcolumns}={}
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolssep}
16 \newlength{\multicolbaselineskip}
17 \newlength{\multicoltolerance}
18 \newlength{\multicolpretolerance}
19 \newcommand*{\columnseprulecolor}{\normalcolor}
20 \newcounter{columnbadness}
21 \newcounter{finalcolumnbadness}
22 \newcounter{collectmore}
23 \newcounter{unbalance}
24 \newlength{\multicolovershoot}
25 \newlength{\multicolundershoot}

26 \end{warpHTML}

```

File 92 lwarp-multirow.sty

§ 168 Package **multirow**

Pkg **multirow** multirow is emulated during HTML output, and used as-is while inside a `\textrimage`.

In a `\textrimage`, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section [58.17](#) for the print-mode versions.

for HTML output: 1 \LWR@ProvidesPackagePass{multirow}

Remember the print-mode version:

```
2 \LetLtxMacro{\LWR@origmultirow}{\multirow}
```

\LWR@multirowborder Set to `left` or `right` to create a thick border for the cell, for use by `bigdelim`:

```
3 \newcommand{\LWR@multirowborder}{}%
```

§ 168.1 Multirow

\LWR@tabledatamultirowtag [*vpos*] {[*numrows*] [*bigstruts*] [*width*] [*fixup*] {[*text*]}

```
4 \RenewDocumentCommand{\multirow}{O{c} m o m o +m}%
5 {%
6 \LWR@traceinfo{*** multirow #1 #2 #4}%
7 \LWR@maybenewtablerow%
```

Print the start of a new table data cell:

```
8 \LWR@htmlltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
9 \IfValueT{#1}{%
10 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
11 \ifstrequal{#1}{t}{style="vertical-align:top" }{}%
12 }%
```

The left/right border, if given:

```
13 \ifdefvoid{\LWR@multirowborder}{}{%
14 style="border-\LWR@multirowborder: 2px solid black" %
15 }
```

A class adds the column spec and the rule:

```
16 class="td%"
```

Append this column's spec:

```
17 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```
18 \ifthenelse{\equal{\LWR@getexarray{\LWR@midrules}}{\theLWR@tablecolspos}}{Y}{rule}{}%
19 "{}"
```

While printing the text, redefine \\ to generate a new line

```

20 \begingroup \LetLtxMacro{\\"}{\LWR@endofline} #6 \endgroup
21 \LWR@stoppars%
22 \global\boolfalse{\LWR@intabularmetadata}%
23 \renewcommand{\LWR@multirowborder}{}%
24 \LWR@traceinfo{*** multirow done}%
25 }%
```

§ 168.2 Combined multicolumn and multirow

⚠ `\multicolumn & \multirow` l warp does not support 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.

```
\multicolumnrow {{<1:cols>}} {{<2:halign>}} {{<3:vpos>}} {{<4:numrows>}} {{<5:bigstruts>}} {{<6:width>}} {{<7:fixup>}}
{{<8:text>}}
```

26 \RenewDocumentCommand{\multicolumnrow}{m m O{} m O{} m O{} +m}{%

Figure out how many extra HTML columns to add for @ and ! columns:

```
27 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn/multirow tag:

```

28 \begingroup\LetLtxMacro{\\"}{\LWR@endofline}%
29 \LWR@domulticolumn[#3][#4]{\theLWR@tabhtmlcoltotal}{#2}{#8}%
30 \endgroup%
```

Move to the next L^ET_X column:

```

31 \addtocounter{LWR@tablecolspos}{#1}%
32 \addtocounter{LWR@tablecolspos}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```

33 \booltrue{LWR@skipatbang}%
34 }

35 \appto{\LWR@restoreorigformatting}{%
36 \LetLtxMacro{\multirow}{\LWR@origmultirow}%
37 \renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
38 }
```

File 93 **lwarf-nameref.sty**

§ 169 Package **nameref**

Pkg **nameref** nameref is emulated by lwarf.

for HTML output: Discard all options for lwarf-nameref:

```
1 \typeout{Using the lwarf html version of package 'nameref' -- discarding options.}
2 \typeout{    Are not using ProvidesPackage, so that other packages}
3 \typeout{    do not attempt to patch lwarf's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```

File 94 **lwarf-needspace.sty**

§ 170 Package **needspace**

Pkg **needspace** needspace is not used during HTML conversion.

for HTML output: Discard all options for lwarf-needspace:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*\{needspace}[1]{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}
```

File 95 **lwarf-newclude.sty**

§ 171 Package **newclude**

Pkg **newclude** Error if newclude is loaded after lwarf.

for HTML output: Discard all options for lwarf-newclude:

```
1 \LWR@ProvidesPackageDrop{newclude}
2 \LWR@loadbefore{newclude}
```

File 96 **l warp-newunicodechar.sty**

§ 172 Package **newunicodechar**

Pkg newunicodechar Error if newunicodechar is loaded after l warp.

Discard all options for l warp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}
2 \LWR@loadbefore{newunicodechar}

File 97 **l warp-nextpage.sty**

§ 173 Package **nextpage**

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for l warp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}
2 \newcommand{\cleartoevenpage}[1] {}
3 \newcommand{\movetoevenpage}[1] {}
4 \newcommand{\cleartooddpage}[1] {}
5 \newcommand{\movetooddpage}[1] {}

File 98 **l warp-nonumonpart.sty**

§ 174 Package **nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

File 99 **l warp-nopageno.sty**

§ 175 Package **nopageno**

Pkg **nopageno** **nopageno** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

File 100 **l warp-nowidow.sty**

§ 176 Package **nowidow**

Pkg **nowidow** **nowidow** is not used during HTML conversion.

Discard all options for **l warp-nowidow**:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{nowidow}

2 \newcommand*{\nowidow}[1] []
3 \newcommand*{\setnowidow}[1] []

4 \newcommand*{\noclub}[1] []
5 \newcommand*{\setnoclub}[1] []
```

File 101 **l warp-ntheorem.sty**

§ 177 Package **ntheorem**

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg **ntheorem** **ntheorem** is patched for use by **l warp**.

CSS styling of theorems and proofs:

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader<style>

where <theoremstyle> is plain, break, etc.

⚠ **Font control** This conversion is not total. Font control is via css, and the custom L^AT_EX font settings are ignored.

⚠ **Equation numbering** ntheorem has a bug with equation numbering in *AMS* environments when the option `thref` is used. lwarf does not share this bug, so equations with `\split`, etc, are numbered correctly with lwarf's HTML output, but not with the print output. It is recommended to use `cleveref` instead of ntheorem's `thref` option.

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 lwarf emulated `hyperref`.

Some disabled options:

```

1 \DeclareOption{thref}={}
2
3
4 \newbool{LWR@ntheoremmarks}
5 \boolfalse{LWR@ntheoremmarks}
6
7 \DeclareOption{thmmarks}{
8 \booltrue{LWR@ntheoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@ntheoremamsthm}
14 \boolfalse{LWR@ntheoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@ntheoremamsthm}}
17
18
19 \DeclareOption{amsmath}={}
20 \DeclareOption{hyperref}={}
21
22
23 \LWR@ProvidesPackagePass{ntheorem}
```

Storage for the style being used for new theorems.

```
24 \newcommand{\LWR@newtheoremstyle}[1]{plain}
```

Patched to remember the style being used for new theorems:

```

25 \gdef\theoremstyle#1{%
26   \@ifundefined{th@#1}{\@warning
27     {Unknown theoremstyle '#1'. Using 'plain'}%
28     \theorem@style{plain}}
29     \renewcommand{\LWR@newtheoremstyle}{plain}%
30   }%
31   {
32     \theorem@style{#1}
33     \renewcommand{\LWR@newtheoremstyle}{#1}%
34   }
35 }
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

36
37 \gdef\xnithm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}%
40     \expandafter\@ifundefined{c@#1}%
41       {\@definecounter{#1}}%
42     \newctr{#1}[#3]%
43     \expandafter\xdef\csname the#1\endcsname{%
44       \expandafter\noexpand\csname the#3\endcsname \thmcntersep
45       {\noexpand\csname the\the\thmnumbering\endcsname{#1}}%
46     \expandafter\gdef\csname mkheader@#1\endcsname
47       {\csname setparms@#1\endcsname
48         \thm{#1}{#1}{#2}
49       }%
50     \global\@namedef{end#1}{\endtheorem}
51     \AtBeginEnvironment{#1}{\edef{\LWR@thisthmstyle}{\csuse{\LWR@thmstyle#1}}}%
52   \fi
53 }
54
55 \gdef\ynithm#1#2{%
56   \ifthm@tempif
57     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}%
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname the\the\thmnumbering\endcsname{#1}}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
64         \thm{#1}{#1}{#2}
65       }%
66     \global\@namedef{end#1}{\endtheorem}
67     \AtBeginEnvironment{#1}{\edef{\LWR@thisthmstyle}{\csuse{\LWR@thmstyle#1}}}%
```

```

68 \fi
69 }
70
71 \gdef\@othm#1[#2]{%
72   \@ifundefined{c@#2}{\nocounterr{#2}}{%
73     \ifthm@tempif
74       \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}{% new
75         \global\@namedef{the#1}{\nameuse{the#2}}%
76         \expandafter\protected\xdef\csname num@addtheoremline#1\endcsname{%
77           \noexpand\@num@addtheoremline{#1}{#3}}%
78         \expandafter\protected\xdef\csname nonum@addtheoremline#1\endcsname{%
79           \noexpand\@nonum@addtheoremline{#1}{#3}}%
80         \theoremkeyword{#3}}%
81       \expandafter\protected\xdef\csname #1Keyword\endcsname
82         {\the\theoremkeyword}%
83       \expandafter\gdef\csname mkheader@#1\endcsname
84         {\csname setparms@#1\endcsname
85           \atthm{#1}{#2}{#3}
86         }%
87       \global\@namedef{end#1}{\endtheorem}
88       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}{% new
89     \fi}
90 }

```

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

91 \newcommand{\LWR@inctheorem}{%
92 \addtocounter{LWR@thisfloat}{1}%
93 \LWR@stoppars%
94 \LWR@htmntag{a id="autofloat-\arabic{LWR@thisfloat}"{} }\LWR@htmntag{/a}%
95 \LWR@startpars%
96 }

```

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>`.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\@ifundefined{th@#1}{%
99     \expandafter\gdef\csname th@#1\endcsname{%
100       \def\@begintheorem####1####2{%
101 \LWR@forcenewpage}{% new
102 \BlockClass{theorembody#1}\LWR@thisthmstyle}{% new
103 \LWR@inctheorem}{% new

```

```
104 #2}%
105     \def@opargbegintheorem####1####2####3{%
106 \LWR@forcenewpage% new
107 \BlockClass{theorembody#1}\LWR@thisthmstyle% new
108 \LWR@inctheorem% new
109 #3}%
110 }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
114
115 \renewtheoremstyle{plain}%
116   {\item[\hspace{\labelsep} \theorem@headerfont
117     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
118   {\item[\hspace{\labelsep} \theorem@headerfont
119     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
120
121 \renewtheoremstyle{break}%
122   {\item[
123 %   \rlap{\vbox{\hbox{
124     \hspace{\labelsep} \theorem@headerfont
125     \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 %   }\hbox{\strut}}}}
127   ]}%
128   {\item[
129 %   \rlap{\vbox{\hbox{
130     \hspace{\labelsep} \theorem@headerfont
131     \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline
132 %   }\hbox{\strut}}}}
133   ]}%
134
135 \renewtheoremstyle{change}%
136   {\item[\hspace{\labelsep}
137     \theorem@headerfont
138     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
139   {\item[\hspace{\labelsep}
140     \theorem@headerfont
141     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145 %   \rlap{\vbox{\hbox{
146     \hspace{\labelsep} \theorem@headerfont
147     \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148 %   }\hbox{\strut}}}}
149   ]}%
150   {\item[
151 %   \rlap{\vbox{\hbox{
152     \hspace{\labelsep} \theorem@headerfont
153     \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
```

```
154 % } \hbox{\strut}"}}
155   ]}
156
157 \renewtheoremstyle{margin}{%
158   {\item[\hspace{\labelsep}\theorem@headerfont
159     \InlineClass{\theoremheadermargin}{\#2 \qquad \#1\theorem@separator}
160   ]}%
161   {\item[\hspace{\labelsep}\theorem@headerfont
162     \InlineClass{\theoremheadermargin}{\#2 \qquad \#1\ (#3)\theorem@separator}
163   ]}%
164
165 \renewtheoremstyle{marginbreak}{%
166   {\item[\hspace{\labelsep}\theorem@headerfont
167     \InlineClass{\theoremheadermarginbreak}{\#2 \qquad \#1\theorem@separator}\newline
168   ]}%
169   {\item[\hspace{\labelsep}\theorem@headerfont
170     \InlineClass{\theoremheadermarginbreak}{\#2 \qquad \#1\ (#3)\theorem@separator}\newline
171   ]}%
172
173 \renewtheoremstyle{nonumberplain}{%
174   {\item[\theorempartname\hspace{\labelsep}
175     \InlineClass{\theoremheaderplain}{\#1\theorem@separator}]}%
176   {\item[\theorempartname\hspace{\labelsep}
177     \InlineClass{\theoremheaderplain}{\#1\ (#3)\theorem@separator}]}%
178
179 \renewtheoremstyle{nonumberbreak}{%
180   {\item[
181 % \rlap{\vbox{\hbox{
182     \hspace{\labelsep}\theorem@headerfont
183     \InlineClass{\theoremheaderbreak}{\#1\theorem@separator}\newline
184 % }}\hbox{\strut}]}
185   ]}%
186   {\item[
187 % \rlap{\vbox{\hbox{
188     \hspace{\labelsep}\theorem@headerfont
189     \InlineClass{\theoremheaderbreak}{\#1\ (#3)\theorem@separator}\newline
190 % }}\hbox{\strut}]}
191   ]}%
192
193 \renewtheoremstyle{empty}{%
194   {\item[]}%
195   {\item[\theorempartname\hspace{\labelsep}\relax
196     \InlineClass{\theoremheaderplain}{\#3}]}%
197
198 \renewtheoremstyle{emptybreak}{%
199   {\item[]}%
200   {\item[\theorempartname\hspace{\labelsep}\relax
201     \InlineClass{\theoremheaderplain}{\#3}]\ \newline}
```

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

202 \ifbool{LWR@ntheoremamsthm}{}{%
203 % upright text via CSS
204     \newtheoremstyle{plainupright}%
205     {\item[\hspace{\labelsep} \textbf{\textit{theoremlabel}}\textbf{\textit{headerfont}}%
206         \textbf{\textit{InLineClass{theoremlabelplain}\#\#1\ \#\#2\textbf{\textit{theoremseparator}}}}]}%
207     {\item[\hspace{\labelsep} \textbf{\textit{theoremlabel}}\textbf{\textit{headerfont}}%
208         \textbf{\textit{InLineClass{theoremlabelplain}\#\#1\ \#\#2\ (\#\#3)\textbf{\textit{theoremseparator}}}}]}%
209
210 % upright text and small caps header via CSS
211     \newtheoremstyle{nonumberplainuprightsc}%
212     {\item[\textbf{\textit{theoremlabel}}\hspace{\labelsep} \textbf{\textit{headerfont}}\hspace{\labelsep} \textbf{\textit{InLineClass{theoremlabelsc}\#\#1\textbf{\textit{theoremseparator}}}}]}%
213     {\item[\textbf{\textit{theoremlabel}}\hspace{\labelsep} \textbf{\textit{headerfont}}\hspace{\labelsep} \textbf{\textit{InLineClass{theoremlabelsc}\#\#1\ (\#\#3)\textbf{\textit{theoremseparator}}}}]}%
214
215 }
```

The following standard configuration is renewed using the new css:

```

216     \theoremstyle{plainupright}
217     \theorembodyfont{\upshape}
218     \theoremsymbol{\HTML{25A1}}% UTF-8 white box
219     \renewtheorem{Example}{Example}
220     \renewtheorem{example}{Example}
221     \renewtheorem{Beispiel}{Beispiel}
222     \renewtheorem{beispiel}{Beispiel}
223     \renewtheorem{Bemerkung}{Bemerkung}
224     \renewtheorem{bemerkung}{Bemerkung}
225     \renewtheorem{Anmerkung}{Anmerkung}
226     \renewtheorem{anmerkung}{Anmerkung}
227     \renewtheorem{Remark}{Remark}
228     \renewtheorem{remark}{Remark}
229     \renewtheorem{Definition}{Definition}
230     \renewtheorem{definition}{Definition}
231
232     \theoremstyle{nonumberplainuprightsc}
233     \theoremsymbol{\HTML{220E}}% UTF-8 end-of-proof
234     \renewtheorem{Proof}{Proof}
235     \renewtheorem{proof}{Proof}
236     \renewtheorem{Beweis}{Beweis}
237     \renewtheorem{beweis}{Beweis}
238     \qedsymbol{\HTML{220E}}% UTF-8 end-of-proof
239
240     \theoremsymbol{}
241 }% not amsthm
```

Only if the `amsthm` option was given:

```
242 \ifbool{LWR@ntheoremamsthm}{%
243   \gdef\th@plain{%
244     \def\theorem@headerfont{\normalfont\bfseries}\itshape%
245     \def\@begintheorem##1##2{%
246       \LWR@forcenewpage% new
247       \BlockClass{theorembodyplain}% new
248       \LWR@inctheorem% new
249       \item[\hspace{-0.5em}\vphantom{.}\hspace{0.5em}]\labelsep
250     }%
251   }%
252   \InlineClass{theoremheaderplain}{##1\ ##2.}%
253 ]}%
254   \def\@opargbegintheorem##1##2##3{%
255     \LWR@forcenewpage% new
256     \BlockClass{theorembodyplain}% new
257     \LWR@inctheorem% new
258     \item[\hspace{-0.5em}\vphantom{.}\hspace{0.5em}]\labelsep
259   }%
260   \InlineClass{theoremheaderplain}{##1\ ##2\ (###3).}%
261 ]}%
262 %
263 \gdef\th@nonumberplain{%
264   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
265   \def\@begintheorem##1##2{%
266     \LWR@forcenewpage% new
267     \BlockClass{theorembodyplain}% new
268     \LWR@inctheorem% new
269     \item[\hspace{-0.5em}\vphantom{.}\hspace{0.5em}]\labelsep
270   }%
271   \InlineClass{theoremheaderplain}{##1.}%
272 ]}%
273   \def\@opargbegintheorem##1##2##3{%
274     \LWR@forcenewpage% new
275     \BlockClass{theorembodyplain}% new
276     \LWR@inctheorem% new
277     \item[\hspace{-0.5em}\vphantom{.}\hspace{0.5em}]\labelsep
278   }%
279   \InlineClass{theoremheaderplain}{##1\ (###3).}%
280 ]}%
281 %
282 \gdef\th@definition{%
283   \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
284   \def\@begintheorem##1##2{%
285     \LWR@forcenewpage% new
286     \BlockClass{theorembodydefinition}% new
287     \LWR@inctheorem% new
288     \item[\hspace{-0.5em}\vphantom{.}\hspace{0.5em}]\labelsep
289   }%
290   \InlineClass{theoremheaderdefinition}{##1\ ##2.}%
291 ]}%
```

```
292 \def\@opargbegingroup{\%  
293 \LWR@forcenewpage% new  
294 \BlockClass{theorembodydefinition}% new  
295 \LWR@inctheorem% new  
296 \item[\hspace{\labelsep}  
297 % \theorem@headerfont  
298 \InlineClass{theorembodydefinition}{##1\ ##2\ (##3).}  
299 ]}  
300  
301 \gdef\th@nonumberdefinition{\%  
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%  
303 \def\@begintheorem{\%  
304 \LWR@forcenewpage% new  
305 \BlockClass{theorembodydefinition}% new  
306 \LWR@inctheorem% new  
307 \item[\hspace{\labelsep}  
308 % \theorem@headerfont  
309 \InlineClass{theorembodydefinition}{##1.}  
310 ]}  
311 \def\@opargbegingroup{\%  
312 \LWR@forcenewpage% new  
313 \BlockClass{theorembodydefinition}% new  
314 \LWR@inctheorem% new  
315 \item[\hspace{\labelsep}  
316 % \theorem@headerfont  
317 \InlineClass{theorembodydefinition}{##1\ (##3).}  
318 ]}  
319  
320 \gdef\th@remark{\%  
321 \def\theorem@headerfont{\itshape}\normalfont%  
322 \def\@begintheorem{\%  
323 \LWR@forcenewpage% new  
324 \BlockClass{theorembodyremark}% new  
325 \LWR@inctheorem% new  
326 \item[\hspace{\labelsep}  
327 % \theorem@headerfont  
328 \InlineClass{theorembodyremark}{##1\ ##2.}  
329 ]}  
330 \def\@opargbegingroup{\%  
331 \LWR@forcenewpage% new  
332 \BlockClass{theorembodyremark}% new  
333 \LWR@inctheorem% new  
334 \item[\hspace{\labelsep}  
335 % \theorem@headerfont  
336 \InlineClass{theorembodyremark}{##1\ ##2\ (##3).}  
337 ]}  
338  
339 \gdef\th@nonumberremark{\%  
340 \def\theorem@headerfont{\itshape}\normalfont%  
341 \def\@begintheorem{\%
```

```
342 \LWR@forcenewpage% new
343     \BlockClass{theorembodyremark}% new
344     \LWR@inctheorem% new
345     \item[\hspace{\labelsep}
346 % \theorem@headerfont
347     \InlineClass{theoremheaderremark}{##1.}]
348 ]
349 \def\@opargbegingroup{\theorembodyremark}{##1##2##3{%
350 \LWR@forcenewpage% new
351     \BlockClass{theorembodyremark}% new
352     \LWR@inctheorem% new
353     \item[\hspace{\labelsep}
354 % \theorem@headerfont
355     \InlineClass{theoremheaderremark}{##1\ (###3).}]
356 ]
357
358 \gdef\th@proof{%
359     \def\theorem@headerfont{\normalfont\bfseries}\itshape%
360     \def\@begintheorem{\begin{##1}%
361 \LWR@forcenewpage% new
362     \BlockClass{theorembodyproof}% new
363     \LWR@inctheorem% new
364     \item[\hspace{\labelsep}
365 % \theorem@headerfont
366     \InlineClass{theoremheaderproof}{##1.}]
367 ]
368 \def\@opargbegingroup{\theorembodyproof}{##1##2##3{%
369 \LWR@forcenewpage% new
370     \BlockClass{theorembodyproof}% new
371     \LWR@inctheorem% new
372     \item[\hspace{\labelsep}
373 % \theorem@headerfont
374     \InlineClass{theoremheaderproof}{##1\ (###3).}]
375 ]
376
377
378
379 \newcounter{proof}%
380 \if@thmmarks
381     \newcounter{currproofctr}%
382     \newcounter{endproofctr}%
383 \fi
384
385 \gdef\proofSymbol{\openbox}
386
387 \newcommand{\proofname}{Proof}
388
389 \newenvironment{proof}[1][\proofname]{
390     \th@proof
391     \def\theorem@headerfont{\itshape}%
```

```

392     \normalfont
393     \theoremsymbol{\HTMLunicode{220E}}% UTF-8 end-of-proof
394     \@thm{proof}{proof}{#1}
395 }%
396 {\@endtheorem}
397
398 }{}% amsthm option

```

Patched for css:

```

399 \let\LWR@origendtheorem\@endtheorem
400 \renewcommand{\@endtheorem}{%
401 \ifbool{\LWR@ntheoremmarks}{%
402     \ifsetendmark{%
403         \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
404         \setendmarkfalse%
405     }%
406 }{}%
407 \LWR@origendtheorem%
408 \ifbool{\LWR@ntheoremmarks}{\global\setendmarktrue}{%
409 \endBlockClass%
410 }

411 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

```

\thm@thmline {\langle 1: printed type \rangle} {\langle 2: # \rangle} {\langle 3: optional \rangle} {\langle 4: page \rangle}

412 \renewcommand{\thm@@thmline@noname}[4]{%
413 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
414 }
415
416 \renewcommand{\thm@@thmline@name}[4]{%
417 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
418 }

```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```
419 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for css:

```

420 \def\listtheorems#1{
421 \LWR@htmlelementclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%
424 \def\thm@list{#1}\thm@processlist
425 \endgroup

```

```
426 \LWR@htmlelementclassend{nav}{lothm}%
427 }
```

Proof QED symbol:

```
428
429 \newcommand{\qed}{\qquad\the\quedsymbol}
430
431 \AtBeginDocument{
432 \def\openbox{\text{\HTMLunicode{25A1}}}% UTF-8 white box
433 \def\blacksquare{\text{\HTMLunicode{220E}}}% UTF-8 end-of-proof
434 \def\Box{\text{\HTMLunicode{25A1}}}% UTF-8 white box
435 }

\thref {\langle label\rangle}

436 \newcommand*{\thref}[1]{\cref{#1}}
```

File 102 **lwarf-pagenote.sty**

§ 178 Package **pagenote**

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

for HTML output:

```
1 \DeclareOption{page}{}
2 \LWR@ProvidesPackagePass{pagenote}
```

File 103 **lwarf-paralist.sty**

§ 179 Package **paralist**

Pkg paralist paralist is supported with minor changes.

for HTML output:

```
1 \LWR@ProvidesPackagePass{paralist}

2 \AtBeginEnvironment{compactitem}{\LWR@itemizestart}
3 \AtEndEnvironment{compactitem}{\LWR@itemizeend}
4 \AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
5 \AtEndEnvironment{compactenum}{\LWR@enumerateend}
6 \AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
7 \AtEndEnvironment{compactdesc}{\LWR@descriptionend}
8 \def\paradescriptionlabel#1{{\normalfont\textbf{#1}}}
```

File 104 **lwarf-parskip.sty**

§ 180 Package **parskip**

Pkg **parskip** **parskip** is ignored.

for HTML output: Discard all options for lwarf-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

File 105 **lwarf-pdflandscape.sty**

§ 181 Package **pdflandscape**

Pkg **pdflandscape** Emulated.

for HTML output: Discard all options for lwarf-pdflandscape:

```
1 \LWR@ProvidesPackageDrop{pdflandscape}
```

File 106 **lwarf-pdfsync.sty**

§ 182 Package **pdfsync**

Pkg **pdfsync** Emulated.

for HTML output: Discard all options for lwarf-pdfsync:

```
1 \LWR@ProvidesPackageDrop{pdfsync}
```

```
2 \let\pdfsync\relax
3 \let\pdfsyncstart\relax
4 \let\pdfsyncstop\relax
```

File 107 **lwarf-placeins.sty**

§ 183 Package **placeins**

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarf-placeins:

for HTML output: 1 \LWR@ProvidesPackageDrop{placeins}
2 \newcommand*{\FloatBarrier}{}

File 108 **lwarf-prelim2e.sty**

§ 184 Package **prelim2e**

Pkg prelim2e Emulated.

for HTML output: Discard all options for lwarf-prelim2e:

1 \LWR@ProvidesPackageDrop{prelim2e}
2 \newcommand{\PrelimText}{}
3 \newcommand{\PrelimTextStyle}{}
4 \newcommand{\PrelimWords}{}

File 109 **lwarf-ragged2e.sty**

§ 185 Package **ragged2e**

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarf-ragged2e:

for HTML output: 1 \LWR@ProvidesPackageDrop{ragged2e}
2 \newcommand*{\Centering}{\centering}
3 \newcommand*{\RaggedLeft}{\raggedleft}
4 \newcommand*{\RaggedRight}{\raggedright}
5 \newcommand*{\justifying}{}

```

6 \newlength{\CenteringLeftskip}
7 \newlength{\RaggedLeftLeftskip}
8 \newlength{\RaggedRightLeftskip}
9 \newlength{\CenteringRightskip}
10 \newlength{\RaggedLeftRightskip}
11 \newlength{\RaggedRightRightskip}
12 \newlength{\CenteringParfillskip}
13 \newlength{\RaggedLeftParfillskip}
14 \newlength{\RaggedRightParfillskip}
15 \newlength{\JustifyingParfillskip}
16 \newlength{\CenteringParindent}
17 \newlength{\RaggedLeftParindent}
18 \newlength{\RaggedRightParindent}
19 \newlength{\JustifyingParindent}
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{justify}{\justifying}{\endjustifying}

```

File 110 **lwarf-realscripts.sty**

§ 186 Package **realscripts**

Pkg **realscripts** **realscripts** is emulated. See **lwarf.css** for the of class **supsubscript**.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{realscripts}

2 \let\LWR@origtextsuperscript{textsuperscript}
3 \let\LWR@origtextsubscript{textsubscript}
4
5 \RenewDocumentCommand{\textsuperscript}{s}{\LWR@origtextsuperscript}
6 \RenewDocumentCommand{\textsubscript}{s}{\LWR@origtextsubscript}
7
8 \let\realsuperscript{textsuperscript}
9 \let\realsubscript{textsubscript}
10
11 \let\fakesuperscript{textsuperscript}
12 \let\fakesubscript{textsubscript}
13
14 \newlength{\subsupersep}
15
16 \newcommand*{\LWR@realscriptsalign}{}%
17
18 \newcommand*{\LWR@setrealscriptsalign}[1]{%
19 \renewcommand*{\LWR@realscriptsalign}{}%
20 \ifthenelse{\equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{text-align:center;}}{}%
21 \ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{text-align:right;}}{}%

```

```

22 }
23
24 \DeclareDocumentCommand \textsubsuperscript {s O{1} mm} {%
25 \LWR@setrealscriptsalign{#2}%
26 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
27 \textsuperscript{#4}\textsubscript{#3}}%
28 }%
29 }
30
31 \DeclareDocumentCommand \textsupsersubscript {s O{1} mm} {%
32 \LWR@setrealscriptsalign{#2}%
33 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
34 \textsubscript{#4}\textsuperscript{#3}}%
35 }%
36 }

```

File 111 **l warp-relsize.sty**

§ 187 Package **relsize**

Pkg **relsize** **relsize** is patched for use by **l warp**.

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 temporarily restored, and so should work as expected.

⚠ **not small** The **HTML** browser's setting for minimum font size may limit how small the output will be displayed.

for HTML output: 1 \LWR@ProvidesPackagePass{relsize}

```

2 \let\LWR@origrelsize\relsize
3 \LetLtxMacro{\LWR@origlarger}{\larger}
4 \LetLtxMacro{\LWR@origsmaller}{\smaller}
5 \let\LWR@relscale\relscale
6 \LetLtxMacro{\LWR@origtextlarger}{\textlarger}
7 \LetLtxMacro{\LWR@origtextsmaller}{\textsmaller}
8 \let\LWR@textscale\textscale
9
10 \appto{\LWR@restoreorigformatting}{%
11 \let\relsize{\LWR@origrelsize}%
12 \LetLtxMacro{\larger}{\LWR@origlarger}%
13 \LetLtxMacro{\smaller}{\LWR@origsmaller}%

```

```

14 \let\relscale\LWR@relscale%
15 \LetLtxMacro{textlarger}{\LWR@origtextlarger%}
16 \LetLtxMacro{textsmaller}{\LWR@origtextsmaller%}
17 \let{textscale}{\LWR@textscale%}
18 }
19
20 \newcounter{LWR@relsizetemp}
21
22 \renewcommand*{\relsize}[1]{}
23 \renewcommand*{\larger}[1][]{}
24 \renewcommand*{\smaller}[1][]{}
25 \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 112 **l warp-romanbar.sty**§ 188 Package **romanbar**

Pkg romanbar romanbar is patched for use by l warp.

An inline class with an overline and underline is used.

for HTML output: 1 \LWR@ProvidesPackagePass{romanbar}

```

2 \DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
3 \InlineClass[%]
4   text-decoration: overline underline ;
5 ]{romanbar}{#1}%
6 }
```

File 113 **l warp-romanbarpagenumber.sty**

§ 189 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{romanbarpagenumber}

File 114 **l warp-rotating.sty**

§ 190 Package **rotating**

Pkg rotating rotating is emulated during HTML output, and the rotating package is ignored.

All rotations are ignored in HTML output.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{rotating}

2 \LetLtxMacro{\sidewaystable}{\table}
3 \let\endsidewaystable\endtable
4
5 \LetLtxMacro{\sidewaysfigure}{\figure}
6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{}{}
9 \newenvironment*{turn}[1]{}{}
10 \newenvironment*{rotate}[1]{}{}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption
```

File 115 **l warp-rotfloat.sty**

§ 191 Package **rotfloat**

Pkg rotfloat rotfloat is emulated during HTML output, and the rotfloat package is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{rotfloat}
2
3 \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.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}{%
6 {
7   \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
8 }%
9 {
10   \DeclareFloatingEnvironment[fileext=#3]{#1}%
11   \DeclareFloatingEnvironment[fileext=#3]{sideways#1}%
12 }%
13 \csletcs{sideways#1}{#1}%
14 \csletcs{endsideways#1}{end#1}%

```

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.

```

15 \cslet{listof#1s}\relax
16 \cslet{listof#1es}\relax
17 }
```

File 116 **l warp-savetrees.sty**

§ 192 Package **savetrees**

Pkg **savetrees** Emulated.

for HTML output: Discard all options for l warp-savetrees:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

File 117 **l warp-scalefnt.sty**

§ 193 Package **scalefnt**

Pkg **scalefnt** scalefnt is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{scalefnt}

```
2 \DeclareRobustCommand\scalefont[1]{}
```

File 118 **l warp-setspace.sty**

§ 194 Package **setspace**

Pkg **setspace** setspace is not used during HTML conversion.

Discard all options for l warp-setspace:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singespacing}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singespacing}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singespacing*}
17 {
18 \LWR@forcenewpage
19 \BlockClass{singespacing}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{}
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {
```

```
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}
```

File 119 **l warp-shadow.sty**

§ 195 Package **shadow**

Pkg shadow shadow is emulated.

for HTML output: Discard all options for l warp-shadow:

```
1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sboxsep
3 \newdimen\sboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }
```

File 120 **l warp-showidx.sty**

§ 196 Package **showidx**

Pkg showidx showidx is ignored.

for HTML output: Discard all options for l warp-showidx:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

File 121 **l warp-showkeys.sty**

§ 197 Package **showkeys**

Pkg showkeys showkeys is ignored.

for HTML output: Discard all options for l warp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}{}
```

File 122 **l warp-sidecap.sty**

§ 198 Package **sidecap**

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for l warp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

[http://tex.stackexchange.com/questions/45401/
use-the-s-star-argument-with-newdocumentenvironment](http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment)
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCtable}{soo}
3 {\IfValueTF{#3}{\table[#3]}{\table}}
4 {\endtable}
5
6 \ExplSyntaxOn
7 \cs_new:cpx {SCtable*} {\SCtable*}
8 \cs_new_eq:cN {endSCtable*} \endSCtable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}{\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpx {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*[wide]{}{}
```

File 123 **l warp-sidenotes.sty**

§ 199 Package **sidenotes**

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarf.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarf:

Stop paragraph handling while creating the caption:

```
2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4     \LWR@stoppars
5     \captionsetup{style=sidecaption}
6     \IfBooleanTF{#1}
7     { % starred
8         \IfNoValueOrEmptyTF{#2}
9         {\marginnote{\caption*{#4}}}
10        {\marginnote{\caption*{#4}}[#2]}
11    }
12    { % unstarring
13        \IfNoValueOrEmptyTF{#2}
14        {\def\@sidenotes@sidecaption@tof{#4}}
15        {\def\@sidenotes@sidecaption@tof{#2}}
16        \IfNoValueOrEmptyTF{#3}
17        {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18        {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19    }
20    \LWR@startpars
21 }
```

Borrowed from the lwarf version of keyfloat:

```
22 \NewDocumentEnvironment{KFLTsidenotes@marginfloat}{O{-1.2ex} m}
23 {%
24     \LWR@maybeinthisfloat%
25     \LWR@forcenewpage
26     \LWR@stoppars%
27     \LWR@htmlltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
28     \LWR@startpars%
29     \captionsetup{type=#2}%
30 }
31 {
32     \LWR@htmldivclassend{div}
33 }
34
35 \RenewDocumentEnvironment{marginfigure}{o}
36     {\begin{KFLTsidenotes@marginfloat}{figure}}
37     {\end{KFLTsidenotes@marginfloat}}
38
```

```

39 \RenewDocumentEnvironment{margintable}{o}
40   {\begin{KFLT}sidenotes@marginfloat}{table}}
41   {\end{KFLT}sidenotes@marginfloat}}
```

The following were changed by sidenotes, and now are reset back to their l warp-supported originals:

Restoring the definition from the L^AT_EX 2_& article.cls source:

```

42 \renewenvironment{figure*}
43   {@dblfloat{figure}}
44   {\end@dblfloat}
45
46 \renewenvironment{table*}
47   {@dblfloat{table}}
48   {\end@dblfloat}
```

File 124 l warp-siunitx.sty

§ 200 Package **siunitx**

Pkg siunitx siunitx is patched for use by l warp.

⚠ **per-mode** Do not use per-mode=fraction, which cannot be seen by the final pdftotext conversion.

Some units will require that the expression be placed inside math mode.

for HTML output: 1 \LWR@ProvidesPackagePass{siunitx}

```

2 \AtBeginDocument{%
3   \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
4   \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
5   \DeclareSIUnit\elementarycharge{\textit{e}}
6   \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
7   \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
8 }% AtBeginDocument
9
```

File 125 l warp-soul.sty

§ 201 Package **soul**

(Based on original code by MELCHIOR FRANZ.)

Pkg soul Emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{soul}
2 \RequirePackage{xcolor}%
  for \convertcolorspec
```

Storage for the colors to use:

```
3 \newcommand*\{\LWR@soululcolor\}{}%
4
5 \newcommand*\{\LWR@soulstcolor\}{}%
6
7 \% \definecolor{\LWR@soulhlcolordefault}{HTML}{F8E800}
8 \% \newcommand*\{\LWR@soulhlcolor\}{\LWR@soulhlcolordefault}
9 \newcommand*\{\LWR@soulhlcolor\}{}%
```

Basic markup with css:

```
10 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}
11 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}
```

Add colors if not empty:

```
12 \newcommand{\soulcolor}[4]{%
13 \ifcsempty{#2}%
14 {%
15 \InlineClass{#3}{#1}}%
16 {%
17 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
18 \InlineClass{#4: \#\LWR@tempcolor}{#3}{#1}}%
19 }%
20 }
21
22 \newcommand{\ul}[1]{%
23 \LWR@soulcolor{#1}\LWR@soululcolor}{uline}{text-decoration-color}%
24 }
25
26 \newcommand{\st}[1]{%
27 \LWR@soulcolor{#1}\LWR@soulstcolor}{sout}{text-decoration-color}%
28 }
29
30 \newcommand{\hl}[1]{%
31 \LWR@soulcolor{#1}\LWR@soulhlcolor}{highlight}{background-color}%
32 }
```

Nullified:

```
33 \newcommand*\{\soulaccent\}[1]{}%
34 \newcommand*\{\soulregister\}[2]{}%
35 \newcommand{\sloppyword}[1]{#1}
```

```

36 \newcommand*{\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
37 \newcommand*{\resetso}={}
38 \newcommand*{\capsdef}[5] {}
39 \newcommand*{\capsreset}={}
40 \newcommand*{\capssave}[1] {}
41 \newcommand*{\capsselect}[1] {}
42 \newcommand*{\setul}[2] {}
43 \newcommand*{\resetul}={}
44 \newcommand*{\setuldepth}[1] {}
45 \newcommand*{\setuloverlap}[1] {}

```

Set colors:

```

46 \newcommand*{\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
47 \newcommand*{\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
48 \newcommand*{\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}

```

Long versions of the user-level macros:

```

49 \let\textso\so
50 \let\textul\ul
51 \let\texthl\hl
52 \let\textcaps\caps

```

File 126 **lwarp-subfig.sty**

§ 202 Package **subfig**

(Based on original 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:

```

1 \LWR@ProvidesPackagePass{subfig}

\sf@@@subfloat {\langle 1 type\rangle} [\langle 2 lof entry\rangle] [\langle 3 caption\rangle] {\langle 4 contents\rangle}

```

The outer minipage allows side-by-side subfloats with \hfill between.

```

2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}%
4 \LWR@stoppars% new
5   \@ifundefined{FBsc@max}{}{%
6     {\FB@readaux{\let\FBsubheight\relax}}%
7   \tempcpta=\one
8   \ifminipage
9     \tempcpta=z@
10    \else\ifdim \lastskip=z@ \else
11      \tempcpta=tw@
12    \fi\fi
13   \ifmaincaptiontop
14     \sf@top=\sf@nearskip
15     \sf@bottom=\sf@farskip
16   \else
17     \sf@top=\sf@farskip
18     \sf@bottom=\sf@nearskip
19   \fi
20   \leavevmode
21   \setbox\tempboxa \hbox{#4}%
22   \tempdima=\wd\tempboxa
23   \ifundefined{FBsc@max}{}{%
24     \global\advance\Xhsize-\wd\tempboxa
25     \dimen@\ht\tempboxa
26     \advance\dimen@\dp\tempboxa
27     \ifdim\dimen@>\FBso@max
28       \global\FBso@max\dimen@
29     \fi}%
30   \vtop\bgroup
31     \vbox\bgroup
32       \ifcase\tempcpta
33         @minipagefalse
34       \or
35         \vskip\sf@top
36       \or
37         \ifdim \lastskip=z@ \else
38           @tempskipb\sf@top\relax\xaddvskip
39         \fi
40       \fi
41       \sf@ifpositiontop{%
42         \ifx \empty#3\relax \else
43           \sf@subcaption{#1}{#2}{#3}%
44           \vskip\sf@capskip
45           \vskip\sf@captionadj
46         \fi\egroup
47         \hrule width0pt height0pt depth0pt
48         \LWR@startpars% new
49 } \box\tempboxa

```

```

50          #4
51          \LWR@stoppars% new
52      }%
53      \LWR@startpars% new
54      \@ifundefined{FBsc@max}%
55      {
56% \box\@tempboxa
57          #4
58      }%
59      {\ifx\FBsuboheight\relax
60%         \box\@tempboxa
61          #4
62      \else
63%         \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
64          #4
65      \fi}%
66      \LWR@stoppars% new
67      \egroup
68      \ifx \empty#3\relax \else
69          \vskip\sf@capskip
70          \hrule width0pt height0pt depth0pt
71          \sf@subcaption{\#1}{\#2}{\#3}%
72      \fi
73  }%
74      \vskip\sf@bottom
75  \egroup
76  \@ifundefined{FBsc@max}{}%
77  {\addtocounter{FRobj}{-1}%
78  \ifnum\c@FRobj=0\else
79      \subfloatrowsep
80  \fi}%
81  \ifmaincaptiontop\else
82      \global\advance\@nameuse{c@\@captype}\m@ne
83  \fi
84 \end{minipage}%
85 \LWR@startpars% new
86 \endgroup\ignorespaces%
87 }%

```

\sf@subcaption {\i type} {\i lofentry} {\i caption}

```

88 \long\def\sf@subcaption#1#2#3{%
89 \LWR@stoppars% new
90 \ifx \relax#2\relax \else
91 \bgroup
92 \let\label=\gobble
93 \let\protect=\string
94 \def\@subcaplabel{%
95 \caption@lstfmt{\@nameuse{p@\#1}}{\@nameuse{the\#1}}}%
96 \sf@updatecaptionlist{\#1}{\#2}{\the\value{\@captype}}{\the\value{\#1}}%

```

```
97      \egroup
98  \fi
99  \bgroup
100    \ifx \relax#3\relax
101      \let\captionlabelsep=\relax
102    \fi
103 %
104 %       \setbox0\vbox{%
105 %
106 %         \hss
107 %         \parbox[t]{\the\@tempdima}{%
108 %
109 %           \caption@make
110 %             {\@nameuse{sub\@capttype name}}%
111 %             {\@nameuse{thesub\@capttype}}%
112 %             {#3}
113 %         }%
114 %
115 %
116     \@ifundefined{FBsc@max}%
117 %       {\box0}%
118 %
119 %   \parbox[t]{\the\@tempdima}{%
120 \LWR@traceinfo{sfsubcap B1}%
121     \LWR@htmlblocktag{figcaption}%
122     \caption@make
123       {\@nameuse{sub\@capttype name}}%
124       {\@nameuse{thesub\@capttype}}%
125       {#3}
126     \LWR@htmlblocktag{/figcaption}%
127 \LWR@traceinfo{sfsubcap B2}%
128 %
129 }%
130 %
131 { \dimen@\ht0%
132   \advance\dimen@\dp0%
133   \ifdim\dimen@>\FBsc@max
134     \global\FBsc@max\dimen@
135   \fi
136   \FB@readaux{\let\FBsubcheight\relax}%
137   \ifx\FBsubcheight\relax
138     \def\next{%
139       \parbox[t]{\the\@tempdima}%
140     }%
141     \else
142     \def\next{%
143       \parbox[t][\FBsubcheight][t]{\the\@tempdima}%
144     }%
145     \vbox{%
146       \hb@xt@{\the\@tempdima}{%
```

```
147
148 %           \hss
149 %           \next{%
150 \LWR@traceinfo{sfsubcap C1}{ new
151           \caption@make
152               {\@nameuse{sub@\capttype name}}%
153               {\@nameuse{thesub@\capttype}}%
154               {#3}
155 \LWR@traceinfo{sfsubcap C1}{ new
156 %   }%
157 %           \hss
158
159 %   }
160       }
161   }%
162 \egroup
163 \LWR@startpars{ new
164 }

\caption@@make {\i<caption label>} {\i<caption text>}

165 \renewcommand\caption@@make[2]{%
166 \LWR@startpars{ new
167   \sbox{\tempboxa{#1}}%
168   \ifdim\wd\tempboxa=\z@
169     \let\caption@lsep\relax
170   \fi
171   \caption@ifempty{#2}{%
172     \let\caption@lsep\empty
173     \let\caption@tfmt\firstofone
174   }%
175 %   \setpar{\@par\caption@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}{ new
177   \caption@applyfont
178   \caption@fmt
179   {\ifcaption@star\else
180     \begingroup
181       \captionlabelfont
182       #1%
183     \endgroup
184   \fi}%
185   {\ifcaption@star\else
186     \begingroup
187       \caption@iflf\captionlabelfont
188       \relax\caption@lsep
189     \endgroup
190   \fi}%
191   {\captiontextfont
192     \caption@ifstrut
193     {\vrule\@height\ht\strutbox\@width\z@}%
```

```

194      {}%
195      \nobreak\hskip\z@skip % enable hyphenation
196      \caption@tfmt{#2}
197      \LWR@ensuredoingapar% new
198      \caption@ifstrut
199      {\ifhmode\@finalstrut\strutbox\fi}%
200      {}%
201      \par}%
202 \LWR@stoppars% new
203 }

```

Patches for \sf@sub@label:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar% new
206   \@ifnextchar(%    ) match left parenthesis
207   { \sf@sub@label}
208   { \sf@sub@label(Sub\@capttype\space
209     \ifundefined{thechapter}{}{\nameuse{thechapter}\space}%
210     \nameuse{p@sub\@capttype}%
211     \nameuse{thesub\@capttype}.)}}}

```

Patches for \subref.

The unstarred version uses a \ref link whose printed text comes from the sub@<label>:

```

212 \renewcommand{\sf@subref}[1]{%
213 \LWR@subnewref{#1}{sub@#1}%
214 }

```

The starred version uses the printed sub@<label> which is stored as if it were a page number:

```
215 \renewcommand{\sf@@subref}[1]{\LWR@origpageref{sub@#1}}
```

Defining new subfloats. The l@sub<type> for each is redefined.

```

216 \LetLtxMacro{\LWR@orig@newsubfloat}{\newsubfloat}
217
218 \def\@newsubfloat[#1]{%
219 \LWR@orig@newsubfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}%
221 }

```

Pre-defined for figures and tables:

```

222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
224 % \def\subfigure{\subfloat}

```

```
225 \% \def\subtable{\subfloat}
```

File 127 **l warp-supertabular.sty**

§ 203 Package **supertabular**

Pkg supertabular supertabular is emulated during HTML output.

for HTML output: 1 \LWR@ProvidesPackageDrop{supertabular}

⚠ misplaced alignment
alignment tab character & For \tablefirsthead, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 7.7.

```
2 \newcommand{\LWRST@firsthead}(){}
3
4 \newcommand{\tablefirsthead}[1]{%
5   \long\gdef\LWRST@firsthead{\#1}%
6 }
7
8 \newcommand{\tablehead}[1]{}
9 \newcommand{\tabletail}[1]{}
10
11 \newcommand{\LWRST@lasttail}{}
12
13 \newcommand{\tablelasttail}[1]{%
14   \long\gdef\LWRST@lasttail{\#1}%
15 }
16
17 \newcommand{\tablecaption}[2][]{%
18   \long\gdef\LWRST@caption{\caption[\#1]{\#2}}%
19 }
20
21 \let\topcaption\tablecaption
22 \let\bottomcaption\tablecaption
23
24 \global\let\LWRST@caption\relax
25
26 \newcommand*\shrinkheight[1]{}
27
28 \NewDocumentEnvironment{supertabular}{s o m}
29 {
```

```

30 \LWR@traceinfo{supertabular}
31 \table
32 \LWRST@caption
33 \begin{tabular}{#3}
34 \TabularMacro\ifdefvoid{\LWRST@firsthead}%
35 {\LWR@getmynexttoken}%
36 {\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
37 }
38 {%
39 \ifdefvoid{\LWRST@lasttail}%
40 {}%
41 {%
42 \TabularMacro\ResumeTabular%
43 \LWRST@lasttail%
44 }%
45 \end{tabular}
46 \endtable
47 \LWR@traceinfo{supertabular done}
48 }
49
50 \NewDocumentEnvironment{mpsuptabular}{s o m}
51 {\minipage{\linewidth}\supertabular{#3}}
52 {\endsupertabular\endminipage}

```

File 128 **l warp-syntonly.sty**

§ 204 Package **syntonly**

Pkg **syntonly** Emulated.

for HTML output: Discard all options for l warp-syntonly:

```

1 \LWR@ProvidesPackageDrop{syntonly}

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*\syntonly{}%
6
7 \onlypreamble\syntonly

```

File 129 **l warp-tabularx.sty**

§ 205 Package **tabularx**

Pkg **tabularx** tabularx is emulated by l warp.

for HTML output: Discard all options for l warp-tabularx:

```
1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
```

File 130 **l warp-tabulary.sty**

§ 206 Package **tabulary**

Pkg **tabulary** tabulary is emulated by l warp.

for HTML output: Discard all options for l warp-tabulary.

Column types L, C, R, and J are emulated by l warp core code.

```
1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

File 131 **lwarp-textarea.sty**

§ 207 Package **textarea**

Pkg **textarea** textarea is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textarea}

2 \newcommand\StartFromTextArea{}
3 \newcommand\StartFromHeaderArea{}
4 \newcommand*\RestoreTextArea{}
5 \newcommand*\ExpandTextArea[1][*]{}
6 \let\NCC@restoretextarea\empty
```

File 132 **lwarp-textcomp.sty**

§ 208 Package **textcomp**

Pkg **textcomp** textcomp is patched for use by lwarp.

⚠ **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.

for HTML output:

```
1 \LWR@ProvidesPackagePass{textcomp}
```

The following are restored for print when inside a `lateximage`:

```

2 \let\LWR@origtextdegree\textdegree
3 \let\LWR@origtextcelsius\textcelsius
4 \let\LWR@origtextohm\textohm
5 \let\LWR@origtextmu\textmu
6 \LetLtxMacro{\LWR@origcapitalcedilla}{\capitalcedilla}
7 \LetLtxMacro{\LWR@origcapitalogonek}{\capitalogonek}
8 \LetLtxMacro{\LWR@origcapitalgrave}{\capitalgrave}
9 \LetLtxMacro{\LWR@origcapitalacute}{\capitalacute}
10 \LetLtxMacro{\LWR@origcapitalcircumflex}{\capitalcircumflex}
11 \LetLtxMacro{\LWR@origcapitaltilde}{\capitaltilde}
12 \LetLtxMacro{\LWR@origcapitaldieresis}{\capitaldieresis}
13 \LetLtxMacro{\LWR@origcapitalhungarumlaut}{\capitalhungarumlaut}
14 \LetLtxMacro{\LWR@origcapitalring}{\capitalring}
15 \LetLtxMacro{\LWR@origcapitalcaron}{\capitalcaron}
16 \LetLtxMacro{\LWR@origcapitalbreve}{\capitalbreve}
```

```

17 \LetLtxMacro{\LWR@origcapitalmacron}{\capitalmacron}
18 \LetLtxMacro{\LWR@origcapitaldotaccent}{\capitaldotaccent}
19 \LetLtxMacro{\LWR@origtextcircled}{\textcircled}

```

For HTML, use HTML entities or direct Unicode, depending on the engine:

```

20 \ifPDFTeX
21 \renewcommand*{\textdegree}{\HTMLentity{deg}}
22 \renewcommand*{\textcelsius}{\HTMLentity{deg}\text{,C}}
23 \renewcommand*{\textohm}{\HTMLunicode{2126}}
24 \renewcommand*{\textmu}{\HTMLunicode{00B5}}
25 \else
26 \renewcommand*{\textdegree}{\text{°}}
27 \renewcommand*{\textcelsius}{\text{°C}}
28 \renewcommand*{\textohm}{\text{Ω}}
29 \renewcommand*{\textmu}{\text{μ}}
30 \fi

```

For HTML, Unicode diacritical marks are used:

```

31 \renewcommand*{\capitalcedilla}[1]{\HTMLunicode{0327}}
32 \renewcommand*{\capitalogonek}[1]{\HTMLunicode{0328}}
33 \renewcommand*{\capitalgrave}[1]{\HTMLunicode{0300}}
34 \renewcommand*{\capitalacute}[1]{\HTMLunicode{0301}}
35 \renewcommand*{\capitalcircumflex}[1]{\HTMLunicode{0302}}
36 \renewcommand*{\capitaltilde}[1]{\HTMLunicode{0303}}
37 \renewcommand*{\capitaldieresis}[1]{\HTMLunicode{0308}}
38 \renewcommand*{\capitalhungarumlaut}[1]{\HTMLunicode{30B}}
39 \renewcommand*{\capitalring}[1]{\HTMLunicode{30A}}
40 \renewcommand*{\capitalcaron}[1]{\HTMLunicode{30C}}
41 \renewcommand*{\capitalbreve}[1]{\HTMLunicode{306}}
42 \renewcommand*{\capitalmacron}[1]{\HTMLunicode{304}}
43 \renewcommand*{\capitaldotaccent}[1]{\HTMLunicode{307}}

```

\textcircled becomes a span with a rounded border:

```

44 \renewcommand*{\textcircled}[1]{%
45 \InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{#1}%
46 }

```

When a `lateXimage` is begun:

```

47 \appto{\LWR@restoreorigformatting}{%
48 \let\textdegree{\LWR@origtextdegree}%
49 \let\textcelsius{\LWR@origtextcelsius}%
50 \let\textohm{\LWR@origtextohm}%
51 \let\textmu{\LWR@origtextmu}%
52 \LetLtxMacro{\capitalcedilla}{\LWR@origcapitalcedilla}%
53 \LetLtxMacro{\capitalogonek}{\LWR@origcapitalogonek}%

```

```

54 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
55 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
56 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
57 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
58 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
59 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
60 \LetLtxMacro\capitalring\LWR@origcapitalring%
61 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
62 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
63 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
64 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
65 \LetLtxMacro\textcircled\LWR@origtextcircled%
66 }

```

File 133 **l warp-textpos.sty**

§ 209 Package **textpos**

Pkg **textpos** **textpos** is emulated during HTML output, and the **textpos** package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r()}{\begin{#1}{}}
3 \NewDocumentEnvironment{textblock*}{m o r()}{\begin{#1}{}}
4 \newcommand*\{\TPGrid}[3]{\begin{#1}\begin{#2}\begin{#3}\end{#3}\end{#2}\end{#1}}
5 \NewDocumentCommand{\TPMargin}{s o}{\begin{#1}\begin{#2}\begin{#3}\end{#3}\end{#2}\end{#1}}
6 \newcommand*\{\textblockcolour}[1]{\color{#1}}
7 \newcommand*\{\textblockrulecolour}[1]{\color{#1}}
8 \newcommand*\{\textblockcolor}[1]{\color{#1}}
9 \newcommand*\{\textblockrulecolor}[1]{\color{#1}}
10 \newcommand*\{\tekstblokkulur}[1]{\color{#1}}
11 \newcommand*\{\tekstblokrulekulur}[1]{\color{#1}}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}[1]{#1}
16 \newcommand*\{\showtextsize}{}
17 \newcommand{\textblockorigin}[2]{#1}

```

File 134 **l warp-theorem.sty**

§ 210 Package **theorem**

(Based on original code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by l warp.

CSS styling of theorems and proofs:

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

Storage for the style being used for new theorems:

2 \newcommand{\LWR@newtheoremstyle}{plain}

Patched to remember the style being used for new theorems:

```

3 \gdef\theoremstyle#1{%
4   \@ifundefined{th@#1}{\@warning
5     {Unknown theoremstyle '#1'. Using 'plain'}%
6     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% new
12  }%
13  \begingroup
14    \csname th@\the\theorem@style \endcsname
15  \endgroup

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

16 \gdef\xnithm#1#2[#3]{%
17   \expandafter\@if definable\csname #1\endcsname
18   {%
19    \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
20    \@definecounter{#1}\@newctr{#1}[#3]%
21    \expandafter\xdef\csname the#1\endcsname
22    {\expandafter\noexpand\csname the#3\endcsname
23     \@thmcOUNTERsep \@thmcOUNTER{#1}}%
24    \def\@tempa{\global\@namedef{#1}}%
25    \expandafter\@tempa \expandafter{%
26      \csname th@\the\theorem@style
27      \expandafter\endcsname \the\theorem@bodyfont
28      \@thm{#1}{#2}}%

```

```

29      \global \expandafter \let \csname end#1\endcsname \endtheorem
30      \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}% new
31    }
32
33 \gdef\@ynthm#1#2{%
34     \expandafter\@if definable\csname #1\endcsname
35   {
36     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37     \@definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcnter{#1}}%
39     \def\@tempa{\global\@namedef{#1}}\expandafter \atempa
40     \expandafter{\csname th@\the \theorem@style \expandafter
41     \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42     \global \expandafter \let \csname end#1\endcsname \endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}% new
44   }
45
46 \gdef\@othm#1[#2]#3{%
47   \expandafter\ifx\csname c@#2\endcsname\relax
48   \nocounterr{#2}%
49   \else
50   \expandafter\@if definable\csname #1\endcsname
51   {
52     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
53     \expandafter \xdef \csname the#1\endcsname
54     {\expandafter \noexpand \csname the#2\endcsname}%
55     \def\@tempa{\global\@namedef{#1}}\expandafter \atempa
56     \expandafter{\csname th@\the \theorem@style \expandafter
57     \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58     \global \expandafter \let \csname end#1\endcsname \endtheorem
59     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}% new
60   }%
61   \fi}

```

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`.

```

62 \gdef\th@plain{\%normalfont\itshape
63   \def\@begintheorem##1##2{%
64 \LWR@forcenewpage% new
65   \BlockClass{theorembody}\LWR@thisthmstyle% new
66     \item[\hskip\labelsep
67       \InlineClass{theoremheader}{##1\ ##2}
68     ]}%

```

```
69 \def\@opargbegintheorem##1##2##3{%
70 \LWR@forcenewpage% new
71     \BlockClass{theorembody}\LWR@thisthmstyle)% new
72     \item[\hspace{\labelsep}
73         \InlineClass{theoremheader}{##1\ ##2\ (##3)}%
74     ]}%
75 }
76
77 \gdef\th@break{\normalfont\slshape
78 \def\@begintheorem##1##2{%
79 \LWR@forcenewpage% new
80     \BlockClass{theorembody}\LWR@thisthmstyle)% new
81     \item[\hspace{\labelsep}
82         \InlineClass{theoremheader}{##1\ ##2}\newline%
83     ]}%
84 \def\@opargbegintheorem##1##2##3{%
85 \LWR@forcenewpage% new
86     \BlockClass{theorembody}\LWR@thisthmstyle)% new
87     \item[\hspace{\labelsep}
88         \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
89     ]}%
90 }
91
92 \gdef\th@marginbreak{\normalfont\slshape
93 \def\@begintheorem##1##2{%
94 \LWR@forcenewpage% new
95     \BlockClass{theorembody}\LWR@thisthmstyle)% new
96     \item[\hspace{\labelsep}%
97         \InlineClass{theoremheader}{##2 \quad ##1}\newline
98     ]}%
99 \def\@opargbegintheorem##1##2##3{%
100 \LWR@forcenewpage% new
101     \BlockClass{theorembody}\LWR@thisthmstyle)% new
102     \item[\hspace{\labelsep}%
103         \InlineClass{theoremheader}{##2 \quad ##1\ %
104             (##3)}\newline
105     ]}%
106 }
107
108 \gdef\th@changebreak{\normalfont\slshape
109 \def\@begintheorem##1##2{%
110 \LWR@forcenewpage% new
111     \BlockClass{theorembody}\LWR@thisthmstyle)% new
112     \item[\hspace{\labelsep}
113         \InlineClass{theoremheader}{##2\ ##1}\newline
114     ]}%
115 \def\@opargbegintheorem##1##2##3{%
116 \LWR@forcenewpage% new
117     \BlockClass{theorembody}\LWR@thisthmstyle)% new
118     \item[\hspace{\labelsep}
```

```

119      \InlineClass{theoremheader}{ ##2\ ##1\ %
120      (##3)}\newline
121  ]}
122 }
123
124 \gdef\th@change{\%\normalfont\slshape
125   \def\@begintheorem##1##2{
126 \LWR@forcenewpage% new
127   \BlockClass{theorembody}\LWR@thisthmstyle}% new
128   \item[\hspace{\labelsep}
129     \InlineClass{theoremheader}{##2\ ##1}
130   ]}%
131 \def\@opargbegintheorem##1##2##3{%
132 \LWR@forcenewpage% new
133   \BlockClass{theorembody}\LWR@thisthmstyle}% new
134   \item[\hspace{\labelsep}
135     \InlineClass{theoremheader}{##2\ ##1\ (##3)}
136   ]}%
137 }
138
139 \gdef\th@margin{\%\normalfont\slshape
140   \def\@begintheorem##1##2{
141 \LWR@forcenewpage% new
142   \BlockClass{theorembody}\LWR@thisthmstyle}% new
143   \item[\hspace{\labelsep}
144     \InlineClass{theoremheader}{##2 \qquad ##1}
145   ]}%
146 \def\@opargbegintheorem##1##2##3{%
147 \LWR@forcenewpage% new
148   \BlockClass{theorembody}\LWR@thisthmstyle}% new
149   \item[\hspace{\labelsep}
150     \InlineClass{theoremheader}{##2 \qquad ##1\ (##3)}
151   ]}%
152 }

```

Patched for css:

```
153 \gdef\@endtheorem{\endBlockClass\endtrivlist}
```

File 135 **lwarf-threeparttable.sty**

§ 211 Package **threeparttable**

Pkg **threeparttable** threeparttable is emulated during HTML output, and the threeparttable package is ignored.

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 of class tnoteitemheader, otherwise they are plain text.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{threeparttable}

Prints the table note item header inside a css class of tnoteitemheader.

2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}


To emulate threeparttable:

3 \newenvironment*{threeparttable}[1][b]{}{}

4 \newenvironment*{tablenotes}[1] []
5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}%
8 \ltx@ifpackageloaded{enumitem}{%
9 \setlist[description]{format=\LWR@printtablenote}%
10 }{}%
11 \description%
12 }%
13 {%
14 \enddescription%
15 \endBlockClass%
16 }

17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}

```

File 136 lwarf-tikz.sty

§ 212 Package **tikz**

Pkg **tikz** tikz is supported.

Accept all options for lwarf-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

catcodes lwarf changes the catcode of \$ for its own use. The Tikz babel library temporarily changes catcodes back to normal for Tikz's use. tikz v3.0.0 introduced the babel library which handles catcode changes. For older versions, lwarf must change \$'s catcode itself.

Also see:

<https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded>

for HTML output:

```

2 \begin{warpHTML}

3 \newboolean{LWR@tikzbabel}
4
5 \@ifpackagelater{tikz}{2013/12/20}%
6 {\use tikzlibrary[babel]\booltrue{LWR@tikzbabel}}
7 {\boolfalse{LWR@tikzbabel}}
```

Env `tikzpicture` `tikzpicture` environment is enclosed inside a `\lateximage`. May be used as-is, and its contents will be converted to an image.

```

8 \BeforeBeginEnvironment{tikzpicture}{%
9 \lateximage%
10 \ifbool{LWR@tikzbabel}%
11 {}%
12 {\catcode`\$=3} % dollar sign is math shift
13 }
14
15 \AfterEndEnvironment{tikzpicture}{%
16 \end lateximage%
17 \ifbool{LWR@tikzbabel}%
18 {}%
19 {\catcode`\$=\active}%
20 }
```

21 \end{warpHTML}

File 137 `l warp-titleps.sty`

§ 213 Package **titleps**

Pkg `titleps` `titleps` is loaded and used by `l warp` during HTML output. All user options and macros are ignored and disabled.

Discard all options for `l warp-titleps`:

for HTML output:

1 \LWR@ProvidesPackageDrop{titleps}

`\pagestyle` and `\thispagestyle` are already disabled in the `l warp` code.

```

2 \NewDocumentCommand{\newpagestyle}{m o m}{}%
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}%
```

```
4 \NewDocumentCommand{\sethead}{o o o m m m}{}  
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}  
  
6 \NewDocumentCommand{\settitemarks}{s m}{}  
  
7 \newcommand*{\headrule}{}  
8 \newcommand*{\footrule}{}  
  
9 \newcommand*{\setheadrule}[1]{}  
10 \newcommand*{\setfootrule}[1]{}  
  
11 \newcommand*{\makeheadrule}{}  
12 \newcommand*{\makefootrule}{}  
  
13 \newcommand{\setmarkboth}[1]{}  
  
14 \NewDocumentCommand{\widenhead}{s o o m m}{}  
  
15 \newcommand*{\bottitemarks}{}  
16 \newcommand*{\toptitemarks}{}  
17 \newcommand*{\firsttitemarks}{}  
18 \newcommand*{\nexttoptitemarks}{}  
19 \newcommand*{\outertitemarks}{}  
20 \newcommand*{\innertitemarks}{}  
  
21 \NewDocumentCommand{\newtitemark}{s m}{}  
  
22 \NewDocumentCommand{\pretitemark}{s m m}{}  
  
23 \newcommand{\ifsamemark}[4]{}  
  
24 \NewDocumentCommand{\setfloatahead}{s o o o m m m m}{}  
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m}{}  
  
26 \NewDocumentCommand{\nextfloatahead}{s o o o m m m m}{}  
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m}{}  
  
28 \newcommand{\newmarkset}[1]{}  
  
29 \NewDocumentCommand{\newextramarkset}{s m m}{}  
  
30 \newcommand{\botextramarks}[1]{}  
31 \newcommand{\topextramarks}[1]{}  
32 \newcommand{\firstextramarks}[1]{}  
33 \newcommand{\nexttopextramarks}[1]{}  
34 \newcommand{\outerextramarks}[1]{}  
35 \newcommand{\innerextramarks}[1]{}  

```

File 138 **l warp-titleref.sty**

§ 214 Package **titleref**

Pkg titleref titleref is superceded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

File 139 **l warp-titlesec.sty**

§ 215 Package **titlesec**

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for l warp-titlesec:

for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}
2 \newcommand*\{\titlelabel}[1]{
3 \newcommand\titleformat{
4 \@ifstar{\ttl@format@s}{
5 {\ttl@format@i}}}
6 \newcommand{\ttl@format@s}[1]{
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{
8 \ifundefined{@chapapp}{\let\@chapapp\chaptername}{
9 \newcommand\chaptitlename{\@chapapp}{
10 \NewDocumentCommand{\titlespacing}{s m m m m o}{
11 \newcommand*\{\filright}{
12 \newcommand*\{\filcenter}{
13 \newcommand*\{\filleft}{
14 \newcommand*\{\fillast}{
15 \newcommand*\{\filinner}{
16 \newcommand*\{\filouter}{
17 \newcommand\wordsep{\fontdimen\tw@\font \oplus
18 \fontdimen\thr@@\font \ominus \fontdimen4\font}
19 \NewDocumentCommand{\titleline}{s o m}{
}

```

20 \providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*{\ttl@rule}[1][]{}
22 \newcommand*{\ttl@row}[2][]{}

23 \newcommand{\iftitlemeasuring}[2]{#2}
24 \newcommand{\assignpagestyle}[2]{#2}
25 \NewDocumentCommand{\titleclass}{m o m o}

```

File 140 **lwarf-titletoc.sty**

§ 216 Package **titletoc**

Pkg **titletoc** titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarf-titletoc:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{titletoc}

2 \NewDocumentCommand{\dottedcontents}{m o m m m}{{}

3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o o}{{}
5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{{}

6 \newcommand{\contentsmargin}[2][]{}

7 \newcommand*{\thecontentslabel}{\thecontentslabel}
8 \newcommand*{\thecontentspage}{\thecontentspage}

9 \newcommand{\contentslabel}[2][]{\thecontentslabel}
10 \newcommand{\contentspage}[1][]{\thecontentspage}

11 \newcommand{\contentspush}[1]{{}

12 \newcommand{\contentsuse}[2]{{}

13 \newcommand*{\startcontents}[1][]{}
14 \newcommand*{\stopcontents}[1][]{}
15 \newcommand*{\resumecontents}[1][]{}

16 \newcommand{\printcontents}[4][]{}

17 \newcommand{\startlist}[2][]{}
18 \newcommand{\stoplist}[2][]{}
19 \newcommand{\resumelist}[2][]{}

```

```
20 \newcommand{\printlist}[4] []{}
```

File 141 **lwarp-titling.sty**

§ 217 Package **titling**

Pkg **titling**

package support lwarp supports the native L^AT_EX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.
⚠ load order

\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 52.7.

The various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

for HTML output: 1 \LWR@ProvidesPackagePass{titling}

\@bsmtitleempty Patch \@bsmtitleempty:

```
2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 }
```

\keepthetitle Patch \keepthetitle:

```
6 \let\LWR@origkeepthetitle\keepthetitle
7 \renewcommand*{\keepthetitle}{%
8 \LWR@orig@keepthetitle%
9 }
```

\killtitle Patch \killtitle:

```
10 \let\LWR@origkilltitle\killtitle
11 \renewcommand*{\killtitle}{%
12 \LWR@orig@killtitle%
13 }
```

Env titlingpage

```
14 \renewenvironment*{titlingpage}
15 {%
```

Start an HTML titlepage div:

```
16 \LWR@printpendingfootnotes
17 \begin{titlepage}
```

Prepare for a custom version of \maketitle inside the titlingpage:

```
18 \LWR@maketitlesetup
19 \let\maketitle\LWR@titlingmaketitle
20 }
21 {
```

At the end of the environment, end the HTML titlepage div:

```
22 \end{titlepage}
23 }
```

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
24
25 \pretitle{}
26 \posttitle{}
27
28 \preauthor{}
29 \postauthor{}
30
31 \predate{}
32 \postdate{}
```

\LWR@maketitlesetup Patches \thanks macros.

```
33 \renewcommand*\LWR@maketitlesetup{}%
```

Redefine the footnote mark:

```
34 \def\@makefnmark{\textsuperscript{\@thefnmark}}
```

```
\thefootnote \Rightarrow \nameuse{arabic}{footnote}, or
\thefootnote \Rightarrow \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
35 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
36 \makethanksmark \LWR@orighspace{1in}
```

```
\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒  
    \@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
37 ##1%  
38 }%  
39 }
```

`\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.

```
40 \renewcommand*\maketitle{%
```

An HTML titlepage <div> is used for all classes.

```
41 \begin{titlepage}
```

Select which kind of footnote marks to use:

```
42 \obsmarkseries
```

Set up special patches:

```
43 \LWR@maketitlesetup
```

Typeset the title, etc:

```
44 \@maketitle
```

Immediately generate any \thanks footnotes:

```
45 \@thanks
```

Close the HTML titlepage div:

```
46 \end{titlepage}
```

Reset the footnote counter:

```
47 \obscontmark  
48 }
```

`\@maketitle` Typesets the title, etc. Patched for HTML.

```
49 \DeclareDocumentCommand{\@maketitle}{}
50     \maketitlehooka
51     {
52         \LWR@stoppars\LWR@htmltag{h1}
53         \obspretitle \@title \obsposttitle
54         \LWR@htmltag{/h1}\LWR@startpars
55     }
56     \maketitlehookb
57     {
58         \begin{BlockClass}{author}
59             \renewcommand{\and}{%
60                 \end{BlockClass}
61                 \begin{BlockClass}{oneauthor}
62             }
63             \begin{BlockClass}{oneauthor}
64                 \obspreauthor \author \obspostauthor
65                 \end{BlockClass}
66                 \end{BlockClass}
67             }
68     \maketitlehookc
69     {
70         \begin{BlockClass}{titledate}
71             \obspredate \date \obspostdate
72             \end{BlockClass}
73     }
74     \maketitlehookd
75 }
```

\LWR@titlingmaketitle \maketitle for use inside an HTML titlingpage environment.

```
76 \renewcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
77 \@thanks
```

Select which kind of footnote marks to use:

```
78 \obsmarkseries
```

Set up special patches:

```
79 \LWR@maketitlesetup
```

Typeset the title, etc:

```
80 \@maketitle
```

Immediately generate any \thanks footnotes:

```
81 \@thanks
```

Reset the footnote counter:

```
82 \@bscontmark
83 }
```

```
\thanksmarkseries {\⟨series⟩}
```

Sets the type of footnote marks used by \thanks, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```
84 \renewcommand{\thanksmarkseries}[1]{%
85 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}}
86 }
```

Set default titlepage thanks footnote marks. See section 52.6.

```
87 \if@titlepage
88   \thanksmarkseries{arabic}
89 \else
90   \thanksmarkseries{fnsymbol}
91 \fi
```

File 142 **lwarf-tocbibind.sty**

§ 218 Package **tocbibind**

Pkg **tocbibind** tocbibind is patched for use by lwarf.

for HTML output: 1 \LWR@ProvidesPackagePass{tocbibind}

```
2 \renewenvironment{theindex}%
3   {%
4     \if@bibchapter
5       \if@donumindex
6         \chapter{\indexname}
7       \else
8         \if@dotocind
9           \chapter*{\indexname}
10          \addcontentsline{toc}{chapter}{\indexname}
11        \else
12          \chapter*{\indexname}
```

```

13          \fi
14          \fi
15      \else
16          \if@donumindex
17              \section{\indexname}
18          \else
19              \if@dotocind
20                  \section*{\indexname}
21                  \addcontentsline{toc}{\@tocextra}{\indexname}
22              \else
23                  \section*{\indexname}
24              \fi
25          \fi
26      \fi
27 \let\item\LWR@indexitem%
28 \let\subitem\LWR@indexsubitem%
29 \let\subsubitem\LWR@indexsubsubitem%
30 }

```

File 143 **lwarf-tocloft.sty**

§ 219 Package **tocloft**

Pkg **tocloft** **tocloft** is emulated. Most user options and macros are ignored and disabled.
`\newlistof` and `\cftchapterprecis` are supported.

Discard all options for **lwarf-tocloft**:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocloft}

2 \newcommand{\tocloftpagestyle}[1]{}

3 \newcommand*{\cftmarktoc}={}
4 \newcommand*{\cfttoctitlefont}={}
5 \newcommand*{\cftaftertoctitle} {}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

8 \newcommand*{\cftmarklof}={}
9 \newcommand*{\cftloftitlefont}={}
10 \newcommand*{\cftafterloftitle} {}

11 \newlength{\cftbeforeloftitleskip}
12 \newlength{\cftafterloftitleskip}

```

```
13 \newcommand*{\cftmarklot}{}
14 \newcommand*{\cftlottitlefont}{}
15 \newcommand*{\cftafterlottitle}{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

18 \newcommand*{\cftdot}{.}
19 \providecommand*{\cftdotsep}{1}
20 \newcommand*{\cftnodots}{5000}
21
22 \providecommand{\cftdotfill}[1]{}

23 \newcommand*{\cftsetpnumwidth}[1]{}
24 \newcommand*{\cftsetrmarg}[1]{}

25 \newcommand*{\cftpnumalign}[1]{}

26 \newlength{\cftparskip}

27 \newlength{\cftbeforepartskip}
28 \newlength{\cftpindent}
29 \newlength{\cftpnumwidth}
30 \newcommand*{\cftpartfont}{}
31 \newcommand*{\cftpartpresnum}{}
32 \newcommand*{\cftpartaftersnum}{}
33 \newcommand*{\cftpartaftersnumb}{}
34 \newcommand*{\cftpartleader}{}
35 \newcommand*{\cftpartdotsep}{1}
36 \newcommand*{\cftpartpagefont}{}
37 \newcommand*{\cftpartafterpnum}{}

38 \newlength{\cftbeforechapskip}
39 \newlength{\cftchapindent}
40 \newlength{\cftchapnumwidth}
41 \newcommand*{\cftchapfont}{}
42 \newcommand*{\cftchappresnum}{}
43 \newcommand*{\cftchapaftersnum}{}
44 \newcommand*{\cftchapaftersnumb}{}
45 \newcommand*{\cftchapleader}{}
46 \newcommand*{\cftchapdotsep}{1}
47 \newcommand*{\cftchappagefont}{}
48 \newcommand*{\cftchapafterpnum}{}

49 \newlength{\cftbeforesecskip}
50 \newlength{\cftsecindent}
51 \newlength{\cftsecnumwidth}
52 \newcommand*{\cftsecfont}{}
```

```
53 \newcommand*{\cftsecpresnum}{}  
54 \newcommand*{\cftsecaftersnum}{}  
55 \newcommand*{\cftsecaftersnumb}{}  
56 \newcommand*{\cftsecleader}{}  
57 \newcommand*{\cftsecdotsep}{1}  
58 \newcommand*{\cftsecpagefont}{}  
59 \newcommand*{\cftsecafterpnum}{}  
  
60 \newlength{\cftbeforesecskip}  
61 \newlength{\cftsubsecindent}  
62 \newlength{\cftsubsecnumwidth}  
63 \newcommand*{\cftsubsecfont}{}  
64 \newcommand*{\cftsubsecpresnum}{}  
65 \newcommand*{\cftsubsecaftersnum}{}  
66 \newcommand*{\cftsubsecaftersnumb}{}  
67 \newcommand*{\cftsubsecleader}{}  
68 \newcommand*{\cftsubsecdotsep}{1}  
69 \newcommand*{\cftsubsecpagefont}{}  
70 \newcommand*{\cftsubsecafterpnum}{}  
  
71 \newlength{\cftbeforesubsecskip}  
72 \newlength{\cftsubsubsecindent}  
73 \newlength{\cftsubsubsecnumwidth}  
74 \newcommand*{\cftsubsubsecfont}{}  
75 \newcommand*{\cftsubsubsecpresnum}{}  
76 \newcommand*{\cftsubsubsecaftersnum}{}  
77 \newcommand*{\cftsubsubsecaftersnumb}{}  
78 \newcommand*{\cftsubsubsecleader}{}  
79 \newcommand*{\cftsubsubsecdotsep}{1}  
80 \newcommand*{\cftsubsubsecpagefont}{}  
81 \newcommand*{\cftsubsubsecafterpnum}{}  
  
82 \newlength{\cftbeforeparaskip}  
83 \newlength{\cftpaiindent}  
84 \newlength{\cftpaiwidth}  
85 \newcommand*{\cftpafont}{}  
86 \newcommand*{\cftparapresnum}{}  
87 \newcommand*{\cftparaftersnum}{}  
88 \newcommand*{\cftparaftersnumb}{}  
89 \newcommand*{\cftparaleader}{}  
90 \newcommand*{\cftparadotsep}{1}  
91 \newcommand*{\cftparapagefont}{}  
92 \newcommand*{\cftparaafterpnum}{}  
  
93 \newlength{\cftbeforeparaskip}  
94 \newlength{\cftsubparaindent}  
95 \newlength{\cftsubparanumwidth}  
96 \newcommand*{\cftsubparafont}{}  
97 \newcommand*{\cftsubparapresnum}{}  
98 \newcommand*{\cftsubparapresnumb}{}  
99 \newcommand*{\cftsubparasep}{1}
```

```
98 \newcommand*{\cftsubparaafersnum}{}  
99 \newcommand*{\cftsubparaafersnumb}{}  
100 \newcommand*{\cftsubparaleader}{}  
101 \newcommand*{\cftsubparadotsep}{1}  
102 \newcommand*{\cftsubparapagefont}{}  
103 \newcommand*{\cftsubparaafterpnum}{}  
  
104 \newlength{\cftbeforefigskip}  
105 \newlength{\cftfigindent}  
106 \newlength{\cftfignumwidth}  
107 \newcommand*{\cftfigfont}{}  
108 \newcommand*{\cftfigpresnum}{}  
109 \newcommand*{\cftfigaftersnum}{}  
110 \newcommand*{\cftfigaftersnumb}{}  
111 \newcommand*{\cftfigleader}{}  
112 \newcommand*{\cftfigdotsep}{1}  
113 \newcommand*{\cftfigpagefont}{}  
114 \newcommand*{\cftfigaftersnum}{}  
  
115 \newlength{\cftbeforesubfigskip}  
116 \newlength{\cftsubfigindent}  
117 \newlength{\cftsubfignumwidth}  
118 \newcommand*{\cftsubfigfont}{}  
119 \newcommand*{\cftsubfigpresnum}{}  
120 \newcommand*{\cftsubfigaftersnum}{}  
121 \newcommand*{\cftsubfigaftersnumb}{}  
122 \newcommand*{\cftsubfigleader}{}  
123 \newcommand*{\cftsubfigdotsep}{1}  
124 \newcommand*{\cftsubfigpagefont}{}  
125 \newcommand*{\cftsubfigaftersnum}{}  
  
126 \newlength{\cftbeforetabskip}  
127 \newlength{\cfttabindent}  
128 \newlength{\cfttabnumwidth}  
129 \newcommand*{\cfttabfont}{}  
130 \newcommand*{\cfttabpresnum}{}  
131 \newcommand*{\cfttabaftersnum}{}  
132 \newcommand*{\cfttabaftersnumb}{}  
133 \newcommand*{\cfttableader}{}  
134 \newcommand*{\cfttabdotsep}{1}  
135 \newcommand*{\cfttabpagefont}{}  
136 \newcommand*{\cfttabaftersnum}{}  
  
137 \newlength{\cftbeforesubtabskip}  
138 \newlength{\cftsubtabindent}  
139 \newlength{\cftsubtabnumwidth}  
140 \newcommand*{\cftsubtabfont}{}  
141 \newcommand*{\cftsubtabpresnum}{}  
142 \newcommand*{\cftsubtabaftersnum}{}  

```

```

143 \newcommand*{\cftsubtabaftersnumb}{}
144 \newcommand*{\cftsubtableader}{}
145 \newcommand*{\cftsubtabdotsep}{1}
146 \newcommand*{\cftsubtabpagefont}{}
147 \newcommand*{\cftsubtabafterpnum}{}

148 \newcommand{\cftsetindents}[3]{}

149 \newcommand{\pagenumbersoff}[1]{}
150 \newcommand{\pagenumberson}[1]{}

```

Emulated through the `\newfloat` mechanism.

```

151 \NewDocumentCommand{\newlistof}{o m m}
152 {%
153 \IfValueTF{#1}
154 {\newfloat{#2}{tbp}{#3}{#1}}
155 {\newfloat{#2}{tbp}{#3}}
156 \cftnamedef{listof#2}{\listof{#2}{#4}}
157 \cftnamedef{#2depth}{1}
158 \expandafter\newlength\csuse{cftbefore#2skip}
159 \expandafter\newlength\csuse{cft#2indent}
160 \expandafter\newlength\csuse{cft#2numwidth}
161 \cftnamedef{cft#2font}{}
162 \cftnamedef{cft#2presnum}{}
163 \cftnamedef{cft#2aftersnum}{}
164 \cftnamedef{cft#2aftersnumb}{}
165 \cftnamedef{cft#2leader}{}
166 \cftnamedef{cft#2dotsep}{1}
167 \cftnamedef{cft#2pagefont}{}
168 \cftnamedef{cft#2afterpnum}{}
169 }

```

`\cftchapterprecis` from `tocloft`:

```

170 \newcommand{\cftchapterprecis}[1]{%
171   \cftchapterprecishere{#1}
172   \cftchapterprecistoc{#1}}
173 \newcommand{\cftchapterprecishere}[1]{%
174   \begin{quote}\textit{#1}\end{quote}}
175 \newcommand{\cftchapterprecistoc}[1]{%
176   \addtocontents{toc}{%
177     \protect\begin{quote}#1\protect\end{quote}}}
178 }
179 }
180 }

```

File 144 **l warp-transparent.sty**

§ 220 Package **transparent**

(Based on original code by HEIKO OBERDIEK.)

Pkg **transparent** Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

 **Not X_ƎT_ƎX!** Note that `transparent` does not work with X_ƎT_ƎX.

for HTML output: Discard all options for `l warp-transparent`:

```

1 \LWR@ProvidesPackageDrop{transparent}

2 \newcommand*{\transparent}[1]{\edef\LWR@opacity{#1}}
3
4 \newcommand*{\texttransparent}[2]{%
5 \begingroup%
6 \transparent{#1}%
7 \InlineClass[opacity: #1]{transparent}{#2}%
8 \endgroup%
9 }
```

File 145 **l warp-trivfloat.sty**

§ 221 Package **trivfloat**

Pkg **trivfloat** `trivfloat` is forced to use the built-in `l warp` emulation for floats.

Discard all options for `l warp-trivfloat`. This tells `trivfloat` not to use `floatrow` or `memoir`.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}
```

for HTML & PRINT:

```
3 \begin{warpall}
```

To create a new float type and change its name:

```

\trivfloat{example}
\renewcommand{\examplename}{Example Name}
```

```
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

4 \end{warpall}

\tfl@chapter@fix Nullified at the beginning of the document. Is used by trivfloat to correct float chapter numbers, but is not needed for lwarf.

for HTML output: 5 \begin{warpHTML}

6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}

7 \end{warpHTML}

§ 221.1 Combining \newfloat, \trivfloat, and algorithmicx

for HTML & PRINT: 8 \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
```

9 \end{warpall}

File 146 lwarf-typearea.sty

§ 222 Package **typearea**

Pkg typearea typearea is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{typearea}

2 \newcommand*\typearea[2][\z@]{}
3 \newcommand*\recalctypearea(){}
4 \@ifundefined{footheight}{\newlength\footheight}{}
5 \newcommand*\areaset[3][\z@]{}

```

File 147 **lwarf-ulem.sty**

§ 223 Package **ulem**

(Based on original code by DONALD ARSENEAU.)

Pkg **ulem** Emulated.

for HTML output: Original lwarf definitions:

```

1 \LetLtxMacro{\LWR@ulemorig}{\emph\emph}
2 \LetLtxMacro{\LWR@ulemorigtextbf}{\textbf\textbf}

```

Basic markup commands, using css:

```

3 \NewDocumentCommand{\uline}{+m}%
4 \InlineClass{uline}{#1}%
5 }%
6
7 \NewDocumentCommand{\uuline}{+m}%
8 \InlineClass{uuline}{#1}%
9 }%
10
11 \NewDocumentCommand{\uwave}{+m}%
12 \InlineClass{uwave}{#1}%
13 }%
14
15 \NewDocumentCommand{\sout}{+m}%
16 \InlineClass{sout}{#1}%
17 }%
18
19 \NewDocumentCommand{\xout}{+m}%
20 \InlineClass{xout}{#1}%
21 }%
22
23 \NewDocumentCommand{\dashuline}{+m}%
24 \InlineClass{dashuline}{#1}%
25 }%
26
27 \NewDocumentCommand{\dotuline}{+m}%

```

```
28 \InlineClass{dotuline}{#1}%
29 }
```

Nullified parameters:

```
30 \NewDocumentCommand{\ULthickness}{}{%
31 \newlength{\ULdepth}}
```

Nullified/emulated macros:

```
32 \NewDocumentCommand{\markoverwith}[m]{}%
33 \NewDocumentCommand{\ULon}[+m]{\uline{#1}\egroup}
```

\useunder only works with \textbf, etc, but not \bfseries, etc.

```
34 \NewDocumentCommand{\useunder}[m m m]{%
35 \relax%
36 \ifx\relax#3\relax\else % argumentative command
37 \def#3{#1}\MakeRobust{#3}\fi
38 }
```

Triggered by package options, also available for the users:

```
39 \newcommand*{\normalem}{\LetLtxMacro{\emph}{\LWR@ulemorigemph}}
40 \newcommand*{\ULforem}{\LetLtxMacro{\emph}{\uline}}
41 \ULforem% default
```

Package options:

```
42 \DeclareOption{normalem}{\normalem}
43 \DeclareOption{ULforem}{\ULforem}
44 \DeclareOption{normalbf}{}
45 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}}
```

Emulate the original package:

```
46 \LWR@ProvidesPackageDrop{ulem}
```

File 148 **l warp-upref.sty**

§ 224 Package **upref**

Pkg upref Ignored.

for HTML output: Discard all options for l warp-upref:

```
1 \LWR@ProvidesPackageDrop{upref}
```

File 149 **l warp-verse.sty**

§ 225 Package **verse**

(Based on original code by PETER WILSON.)

Pkg **verse** **verse** is supported and patched by **l warp**.

for HTML output: Pass all options for **l warp-verse**:

```
1 \LWR@ProvidesPackagePass{verse}
```

\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. **l warp** 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 **\leftskip** These lengths are used by **verse** and **memoir** to control the left margin, and they
Len **\leftmargini** may already be set by the user for print output. New lengths **\HTMLleftskip** and
Len **\TMLvleftskip** **\HTMLleftmargini** are provided to control the margins in **HTML** output. These new
Len **\TMLleftmargini** 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.

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**.

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching verse.}
```

At the beginning of the `verse` environment:

```

4 \AtBeginEnvironment{verse}
5 {%
Pkg  verse   The verse or memoir packages can place stanza numbers to the left with their
Pkg  memoir  \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:
Len  \leftskip 6 \ifdefined{\vleftskip}{%
7 \setlength{\vleftskip}{\HTMLvleftskip}}
8 \setlength{\leftmargini}{\HTMLleftmargini}
9 }{%
10 \LWR@forcenewpage
11 \LWR@atbeginverbatim{verse}
12 \unskip\LWR@origvspace{-\baselineskip}
13 }

```

After the end of the `verse` environment, which places the `pre` tag at the regular left margin:

```

14 \AfterEndEnvironment{verse}{%
15 \unskip\LWR@origvspace{-\baselineskip}
16 \LWR@afterendverbatim
17 }

```

Patch to place `poemtitle` inside an HTML span of class `poemtitle`:

```

18 \ifdef{\poemtitle}{%
19 \DeclareDocumentCommand{\vstypepoemtitle}{m}{%
20   \vspace{\beforepoemtitleskip}%
21   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22   \vspace{\afterpoemtitleskip}%
23 }
24 }{%
25 }% AfterEndPreamble

```

File 150 **lwarp-wallpaper.sty**

§ 226 Package **wallpaper**

Pkg `wallpaper` `wallpaper` is emulated during `HTML` output, and the `wallpaper` package is ignored.
for HTML output: 1 `\LWR@ProvidesPackageDrop{wallpaper}`

```

2 \newcommand*{\CenterWallPaper}[2]{}
3 \newcommand*{\ThisCenterWallPaper}[2]{}
4 \newcommand*{\TileWallPaper}[3]{}
5 \newcommand*{\ThisTileWallPaper}[3]{}
6 \newcommand*{\TileSquareWallPaper}[2]{}
7 \newcommand*{\ThisTileSquareWallPaper}[2]{}
8 \newcommand*{\ULCornerWallPaper}[2]{}
9 \newcommand*{\ThisULCornerWallPaper}[2]{}
10 \newcommand*{\LLCornerWallPaper}[2]{}
11 \newcommand*{\ThisLLCornerWallPaper}[2]{}
12 \newcommand*{\URCornerWallPaper}[2]{}
13 \newcommand*{\ThisURCornerWallPaper}[2]{}
14 \newcommand*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}){}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}

```

File 151 **lwarf-wrapfig.sty**

§ 227 Package **wrapfig**

Pkg **wrapfig** **wrapfig** is emulated during HTML output, and the **wrapfig** package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{wrapfig}

Computed width of a wrapped object. Used to print the **HTML** style.

```

2 \newlength{\LWR@wrapwidth}

3 \newcommand*{\LWR@wrapposition}(){}
4
5 \newcommand*{\LWR@subwrapfigure}[2]{%
6 \LWR@maybeinthisfloat%
7 \renewcommand*{\LWR@wrapposition}{}%
8 \ifthenelse{%
9     \equal{#1}{r}\OR\equal{#1}{R}\OR%
10    \equal{#1}{o}\OR\equal{#1}{O}%
11 }{%
12 \renewcommand*{\LWR@wrapposition}{float:right}%
13 \renewcommand*{\LWR@wrapposition}{float:left}%
14 \setlength{\LWR@wrapwidth}{#2}%
15 \addtolength{\LWR@wrapwidth}{4em}%
16 \uselengthunit{PT}%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmntag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"}

```

```

20 style="width:\rndprintlength{\LWR@wrapwidth} ; %
21 \LWR@wrapposition"%
22 }%
23 \LWR@startpars
24 }
25
26
27 \NewDocumentEnvironment{wrapfigure}{o m o m}
28 {%
29 \LWR@subwrapfigure{#2}{#4}%
30 \captionsetup{type=figure}%
31 }%
32 {%
33 \LWR@htmldivclassend{div}
34 }
35
36
37 \NewDocumentEnvironment{wraptable}{o m o m}
38 {%
39 \LWR@subwrapfigure{#2}{#4}%
40 \captionsetup{type=table}%
41 }%
42 {%
43 \LWR@htmldivclassend{div}
44 }
45
46
47 \NewDocumentEnvironment{wrapfloat}{m o m o m}
48 {%
49 \LWR@subwrapfigure{#3}{#5}%
50 \captionsetup{type=#1}%
51 }%
52 {%
53 \LWR@htmldivclassend{div}
54 }
55
56 \newlength{\wrapoverhang}

```

File 152 **lwarf-xcolor.sty**

§ 228 Package **xcolor**

Pkg **xcolor** **xcolor** is supported by lwarf.

**\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 ****. These **<div>**s are

set to `display: inline-block` so adjacent `\colorboxBlocks` 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.

tables Colored tables are ignored so far. Use CSS to style tables.

colored text and boxes `\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

\color and \pagecolor `\color` and `\pagecolor` are ignored. Use CSS or `\textcolor` where possible.

§ 228.1 **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 redefinitions 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 68.6. 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 324.

\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 lwarf-xcolor, page 538.

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 lwarf-xcolor, page 539.

HTML lateximage: Remembers and reuses the print version.

\colorboxBlock:

Print: Becomes \colorbox.

HTML: Newly defined by lwarf-xcolor to use a <div>, page 539.

HTML lateximage: Remembers and reuses the print version \colorbox.

\fcolorbox:

Print: Modified to allow a background of none.

 \LWRprint@fcolorbox at section 68.6

HTML: Redefined by lwarf-xcolor, page 540.

HTML lateximage: Remembers and reuses the print version.

\fcolorboxBlock:

Print: Becomes \fcolorbox. Section 68.6

HTML: Newly defined by lwarp-xcolor to use a <div>, page 540.

HTML `lateximage`: Remembers and reuses the print version \fcolorbox.

fcolorminipage:

Print: Newly defined in the lwarp core.

LWRprint@fcolorminipage at section 68.6

HTML: Newly defined by lwarp-xcolor, page 541.

HTML `lateximage`: Uses the print version.

\boxframe:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 542.

HTML `lateximage`: Remembers and reuses the print version.

§ 228.2 Code

```
for HTML output: 1 \LWR@ProvidesPackagePass{xcolor}
                  2 \begin{warpHTML}
```

Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```
3 \LetLtxMacro{\LWRprint@textcolor}{\textcolor}
4 \LetLtxMacro{\LWRprint@pagecolor}{\pagecolor}
5 \LetLtxMacro{\LWRprint@nopagecolor}{\nopagecolor}
6 \LetLtxMacro{\LWRprint@colorbox}{\colorbox}
7 \LetLtxMacro{\LWRprint@colorboxBlock}{\colorbox}
8 \LetLtxMacro{\LWRorigprint@fcolorbox}{\fcolorbox}
9 \LetLtxMacro{\LWRorigprint@fcolorboxBlock}{\fcolorbox}
10 \LetLtxMacro{\LWRorigprint@boxframe}{\boxframe}
```

`\LWR@restoreorigformatting` Inside a `lateximage` the following gets restored to their print-mode actions:

```
11 \appto{\LWR@restoreorigformatting}{%
12 \LetLtxMacro{\textcolor}{\LWRprint@textcolor}%
13 \LetLtxMacro{\pagecolor}{\LWRprint@pagecolor}%
14 \LetLtxMacro{\nopagecolor}{\LWRprint@nopagecolor}%
15 \LetLtxMacro{\colorbox}{\LWRprint@colorbox}%
16 \LetLtxMacro{\fcolorbox}{\LWRprint@fcolorbox}%
17 \LetLtxMacro{\boxframe}{\LWRorigprint@boxframe}%
18 }
```

`\LWR@tempcolor` The color converted to HTML colorspace.

```
19 \newcommand*\{\LWR@tempcolor\}{}  
20 \newcommand*\{\LWR@tempcolortwo\}{}  
21 \NewDocumentCommand{\LWR@colorstyle}{m m}{%  
22 \begingroup%  
23 \LWR@FBcancel%
```

`\LWR@colorstyle` {<1: styletext>} {<2: model>} {<3: color>}

For a color style, prints the color converted to HTML colors.

```
24 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Use the `xcolor` package to convert to an HTML color space:

```
25 \#\LWR@tempcolor%
```

Print the converted color:

```
26 \endgroup%  
27 }
```

`\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.

```
28 \newcommand*\{\LWR@borderpadding\}[2]{%  
29 \uselengthunit{PT}%">  
30 border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %  
31 padding:\rndprintlength{\fboxsep}%">  
32 }
```

`\color` `\color` appears in the L^AT_EX PDF output, but is ignored by `pdftotext` and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is `\let` to the final name once the HTML conversion starts.

`\textcolor` [<model>] {<color>} {<text>}

Converted into an HTML hex color span.

```
33 \RenewDocumentCommand{\textcolor}{O{named} m m}{%  
34 \begingroup%  
35 \LWR@FBcancel%
```

```

36 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
37 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
38 \renewcommand*\{\LWR@currenttextcolor}{\#\LWR@tempcolor}%
39 #3%
40 }%
41 \endgroup%
42 }

```

\pagecolor [*model*] {*color*}

Ignored. Use \CSSFilename instead.

```
43 \renewcommand*\{\pagecolor}[2][named]{}
```

\nopagecolor Ignored.

```
44 \let\nopagecolor\relax
```

\colorbox [*model*] {*color*} {*text*}

Converted into an HTML hex background color .

```

45 \RenewDocumentCommand{\colorbox}{O{named} m +m}{%
46 \begingroup%
47 \LWR@FBcancel%
48 \uselengthunit{PT}%
49 \InlineClass[% 
50 background:\LWR@colorstyle{#1}{#2} ; %
51 padding:\rndprintlength{\fboxsep}%
52 ]{\colorbox}{#3}%
53 \endgroup%
54 }

```

\colorboxBlock [*model*] {*color*} {*text*}

Converted into an HTML hex background color <div>.

```

55 \NewDocumentCommand{\colorboxBlock}{O{named} m +m}{%
56 \begingroup%
57 \LWR@FBcancel%
58 \uselengthunit{PT}%
59 \begin{BlockClass}[% 
60 background:\LWR@colorstyle{#1}{#2} ; %
61 padding:\rndprintlength{\fboxsep}%
62 ]{\colorboxBlock}

```

```

63 #3
64 \end{BlockClass}%
65 \endgroup%
66 }

```

\fcolorbox [⟨framemode⟩] {⟨framecolor⟩} [⟨boxmode⟩] {⟨boxcolor⟩} {⟨text⟩}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

67 \RenewDocumentCommand{\fcolorbox}{O{named} m O{named} m +m}{%
68 \LWR@traceinfo{HTML fcolorbox #2 #4}%
69 \begingroup%
70 \LWR@FBcancel%
71 \uselengthunit{PT}%
72 \LWR@forceminwidth{\fboxrule}%
73 \ifthenelse{\equal{#4}{none}}{%
74 {%
75     \InlineClass{%
76         \LWR@borderpadding{#1}{#2}%
77         ]\fcolorbox}{#5}%
78 }%
79 {%
80     \InlineClass{%
81         \LWR@borderpadding{#1}{#2} ; %
82         background:\LWR@colorstyle{#3}{#4}%
83         ]\fcolorbox}{#5}%
84 }%
85 \endgroup%
86 }

```

\fcolorboxBlock [⟨framemode⟩] {⟨framecolor⟩} [⟨boxmode⟩] {⟨boxcolor⟩} {⟨text⟩}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

87 \NewDocumentCommand{\fcolorboxBlock}{O{named} m O{named} m +m}{%
88 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
89 \begingroup%
90 \LWR@FBcancel%
91 \uselengthunit{PT}%
92 \LWR@forceminwidth{\fboxrule}%
93 \ifthenelse{\equal{#4}{none}}{%
94 {%
95     \begin{BlockClass}{%

```

```

96      \LWR@borderpadding{#1}{#2}%
97      ]{fcolorboxBlock}
98      #5
99      \end{BlockClass}%
100 }%
101 {%
102     \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
103     \begin{BlockClass}[%  

104     background:#\LWR@tempcolortwo; %
105     \LWR@borderpadding{#1}{#2}%
106     ]{fcolorboxBlock}
107     #5
108     \end{BlockClass}%
109 }%
110 \endgroup%
111 \LWR@traceinfo{HTML fcolorboxBlock done}%
112 }

```

Env **fcolorminipage** [*1:framemode*] [*2:framecolor*] [*3:boxmode*] [*4:boxcolor*] [*5:align*] [*6:height*] [*7:inner-align*] [*8:width*]

Creates a framed HTML <div> around its contents.

A print-output version is defined in the l warp core: section 68.6

\LWR@subfcolorminipage {[*framemode*} {[*framecolor*} {[*background tag*} {[*height*} }

```

113 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
114 \begin{BlockClass}[%  

115 #3%
116 \LWR@borderpadding{#1}{#2} ; %
117 \IfValueTF{#4}{height:\rndprintlength{\LWR@tempheight} ; }{}%
118 width:\rndprintlength{\LWR@tempwidth}%
119 ]{fcolorminipage}%
120 }

```

fcolorminipage [*1:framestyle*] {[*2:framecolor*} {[*3:bkgcolorstyle*] {[*4:bkgcolor*} {[*5:align*] [*6:height*] [*7:inneralign*] [*8:width*] }

```

121 \NewDocumentEnvironment{fcolorminipage}{O{named} m O{named} m O{c} o o m}
122 {%
123 \LWR@FBcancel%
124 \setlength{\LWR@tempwidth}{#8}%
125 \IfValueTF{#6}{\setlength{\LWR@tempheight}{#6}}{}%
126 \uselengthunit{PT}%
127 \LWR@forceminwidth{\fboxrule}%
128 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
129 \ifthenelse{\equal{#4}{none}}{%

```

```

130 {\LWR@subfcolorminipage{#1}{#2}{ }{#6}}%
131 {%
132     \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
133     \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ; }{#6}%
134 }%
135 }
136 {\end{BlockClass}}

```

\boxframe {*width*} {*height*} {*depth*}

The depth is added to the height, but the box is not descended below by the depth. **\textcolor** is honored.

```

137 \renewcommand*{\boxframe}[3]{%
138 {%
139 \setlength{\LWR@tempwidth}{#1}%
140 \setlength{\LWR@tempheight}{#2}%
141 \addtolength{\LWR@tempheight}{#3}%
142 \uselengthunit{PT}%
143 \LWR@forceminwidth{\fboxrule}%
144 \InlineClass[%
145 display:inline-block ; %
146 border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
147 width:\rndprintlength{\LWR@tempwidth} ; %
148 height:\rndprintlength{\LWR@tempheight}%
149 ]{\boxframe}{}%
150 }%
151 }

152 \end{warpHTML}

```

File 153 **lwarf-xfrac.sty**

§ 229 Package **xfrac**

Pkg **xfrac** Supported by adding **xfrac** instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

⚠ font size In the user's document preamble, **lwarf** should be loaded after font-related setup. During HTML conversion, this font is used by **lwarf** 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@htmlltagc{span style="font-size:\xfracHTMLfontsize"}{}}%
8 \LWR@nestspan%
9 %
10 }
11
12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmlltagc{/span}%
14 \endLWR@nestspan%
15 }
```

\scalebox A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```
16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}
```

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.

```
17 \DeclareInstance{xfrac}{default}{text}{%
18 numerator-format = {%
```

```
19 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
20 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend}%
21 denominator-format = {%
22 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
23 \LWR@htmlsmallfontstart{}#,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
24 scaling = false
25 }
26 \DeclareInstance{xfrac}{lmr}{text}{%
27 numerator-format = {%
28 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
29 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend}%
30 denominator-format = {%
31 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
32 \LWR@htmlsmallfontstart{}#,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
33 scaling = false
34 }
35 \DeclareInstance{xfrac}{lmss}{text}{%
36 numerator-format = {%
37 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
38 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend}%
39 denominator-format = {%
40 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
41 \LWR@htmlsmallfontstart{}#,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
42 scaling = false
43 }
44 \DeclareInstance{xfrac}{lmtt}{text}{%
45 numerator-format = {%
46 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
47 \LWR@htmlsmallfontstart{textsuperscript{#1}\,,\LWR@htmlsmallfontend}%
48 denominator-format = {%
49 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
50 \LWR@htmlsmallfontstart{}#,#1\LWR@htmlsmallfontend},
```

For pdftotext, do not scale the text:

```
51 scaling = false
52 }
53 \end{warpHTML}
```

File 154 **l warp-xltxtra.sty**

§ 230 Package **xltxtra**

Pkg **xltxtra** **xltxtra** is emulated.

(Based on original code by WILL ROBERTSON, JONATHAN KEW.)

for HTML output: 1 \LWR@ProvidesPackageDrop{xltxtra}

```
2 \RequirePackage{realscripts}
3 \RequirePackage{metalogo}
4 \newcommand*\TeX@logo@spacing[6]{}
5
6 \newcommand*{\vfrac}[2]{%
7   \textsuperscript{\#1}/\textsubscript{\#2}%
8 }
9
10 \newcommand\namedglyph[1]{%
11   \tempcnta=\XeTeXglyphindex "#1"\relax
12   \ifnum\tempcnta>0
13     \XeTeXglyph\tempcnta
14   \else
15     \xxt@namedglyph@fallback{\#1}%
16   \fi}
17
18 \newcommand\xxt@namedglyph@fallback[1]{[\#1]}
19
20 \newcommand{\showhyphens}[1]{}
```

File 155 **l warp-xmpincl.sty**

§ 231 Package **xmpincl**

Pkg **xmpincl** Emulated.

for HTML output: Discard all options for **l warp-xmpincl**:

```
1 \LWR@ProvidesPackageDrop{xmpincl}
2 \newcommand*{\includexmp}[1]{}
```

File 156 **lwarp-xtab.sty**

§ 232 Package **xtab**

Pkg **xtab** **xtab** is emulated during HTML output.

for HTML output: `\LWR@ProvidesPackageDrop{xtab}`

⚠ misplaced alignment
alignment tab character & For `\tablefirsthead`, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 7.7.

```
2 \newcommand{\LWRXT@firsthead}{}%
3
4 \newcommand{\tablefirsthead}[1]{%
5   \long\gdef\LWRXT@firsthead{\#1}%
6 }
7
8 \newcommand{\tablehead}[1]{}%
9
10 \newcommand{\tablelasthead}[1]{}%
11
12 \newcommand{\notablelasthead}{}%
13
14 \newcommand{\tabletail}[1]{}%
15
16 \newcommand{\LWRXT@lasttail}{}%
17
18 \newcommand{\tablelasttail}[1]{%
19   \long\gdef\LWRXT@lasttail{\#1}%
20 }
21
22 \newcommand{\tablecaption}[2][]{%
23   \long\gdef\LWRXT@caption{\caption[\#1]{\#2}}%
24 }
25
26 \let\topcaption\tablecaption
27 \let\bottomcaption\tablecaption
28
29 \global\let\LWRXT@caption\relax
30
```

```

31 \newcommand*{\shrinkheight}[1]{}
32
33 \newcommand*{\xentrystretch}[1]{}
34
35 \NewDocumentEnvironment{xtabular}{s o m}
36 {
37 \LWR@traceinfo{xtabular}
38 \table
39 \LWRXT@caption
40 \begin{tabular}{#3}
41 \TabularMacro\ifdefvoid{\LWRXT@firsthead}%
42 {\LWR@getmynexttoken}%
43 {\expandafter\LWR@getmynexttoken\LWRXT@firsthead}%
44 }
45 {%
46 \ifdefvoid{\LWRXT@lasttail}%
47 {}%
48 {%
49 \TabularMacro\ResumeTabular%
50 \LWRXT@lasttail%
51 }%
52 \end{tabular}
53 \endtable
54 \LWR@traceinfo{xtabular done}
55 }
56
57 \NewDocumentEnvironment{mpxtabular}{s o m}
58 {\minipage{\linewidth}\xtabular{#3}}
59 {\endxtabular\endminipage}
```

File 157 **l warp-zwpagelayout.sty**

§ 233 Package **zwpagelayout**

Pkg zwpagelayout zwpagelayout is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{zwpagelayout}

2 \def\noBboxes{}
3 \onlypreamble\noBboxes
4
5 \expandafter\ifx\csname definecolor\endcsname\relax \else
6   \definecolor{cmykblack}{cmyk}{0,0,0,1}
7   \definecolor{grblack}{gray}{0}
8 %   \ifzwpl@redefineblack
9 %     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
10%    \fi
```

```
11 \definecolor{cmykred}{cmyk}{0,1,1,0}
12 \definecolor{cmykgreen}{cmyk}{1,0,1,0}
13 \definecolor{cmykblue}{cmyk}{1,1,0,0}
14 \definecolor{rgbred}{rgb}{1,0,0}
15 \definecolor{rgbgreen}{rgb}{0,1,0}
16 \definecolor{rgbblue}{rgb}{0,0,1}
17 % \ifzwp@redefinetocmyk
18 %   \definecolor{red}{cmyk}{0,1,1,0}
19 %   \definecolor{green}{cmyk}{1,0,1,0}
20 %   \definecolor{blue}{cmyk}{1,1,0,0}
21 % \fi
22 \fi
23
24 \let\OverprintXeTeXExtGState\relax
25
26 \DeclareRobustCommand{\SetOverprint{\ignorespaces}}
27 \DeclareRobustCommand{\SetKnockout{\ignorespaces}}
28 \DeclareRobustCommand{\textoverprint[1]{{\SetOverprint#1}}}
29 \DeclareRobustCommand{\textknockout[1]{{\SetKnockout#1}}}
30
31 \def\SetPDFminorversion#1{}
32 @onlypreamble\SetPDFminorversion
33
34 \newcommand*\Vcorr(){}
35
36 \DeclareRobustCommand{\vb[1] [] {}}
37 \NewDocumentCommand{\NewOddPage}{* o}{}%
38 \NewDocumentCommand{\NewEvenPage}{* o}{}%
39 \def\SetOddPageMessage#1{\gdef\ZW@oddwarning}
40 \def\SetEvenPageMessage#1{\gdef\Z@evenwarning}
41 \def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning
42
43 \def\clap#1{#1}
44
45 \def\CropFlap{2in}
46 \def\CropSpine{1in}
47 \def\CropXSpine{1in}
48 \def\CropXtrim{.25in}
49 \def\CropYtrim{.25in}
50 \def\UserWidth{5in}
51 \def\UserLeftMargin{1in}
52 \def\UserRightMargin{1in}
53 \def\UserTopMargin{1in}
54 \def\UserBotMargin{1in}
55 \def\thePageNumber{\#\!,\arabic{page}}
56 \ifXeTeX
57 \def\ifcaseZWdriver{\ifcase2}
58 \else
59 \def\ifcaseZWdriver{\ifcase1}
60 \fi
```

```
61 \DeclareRobustCommand\ZWifdriver[2]{}
```

Change History and Index

§ 233 Change History

v0.10	General: 2016/03/08 Initial version . . 1	Docs: Table: Float data structures. 304	
v0.11	General: 2016/03/11 1	Docs: Trademarks section. 387	
	Added section: Operating-System portability. 109	Docs: Troubleshooting cross-references. 101	
	Added section: Selecting the operating system. 68	Test Suite: Assigned cleveref name for Test Float. 1	
	Test Suite: limages and index in README.txt 1	Test Suite: Floatrow 1	
	Test Suite: MS-Windows in README.txt 1		
v0.12	\LWR@newhtmlfile: Bugfix: TOC with numbered files. 220	General: 2016/04/06 1	
	General: 2016/03/14 1	Added 426	
	Global: Uses \p@{type} in float captions. 1	Ampersand (&): Fixed handling when passed as an argument. . 264	
	Test Suite: Sub-figures 1	Docs: Added warning icons for items needing special attention. 106	
v0.13	\CaptionSeparator: Fix for newer babel package. 306	Docs: Clarify print/HTML output. 69	
	\LWR@LwarpStart: \up and \fup . . 236	Docs: Moved the supported functions table to the introduction. 34	
	General: 2016/03/24 1	Files: lwarp_formal.css added. . . 1	
	Removed package: subfig 1	Fix: steps counter 426	
	Test Suite: Ordinals, Subcaption . . 1	Fixed & handling. 424	
	tikzpicture: Fix dollar-redefined bug for newer package. 513	Test Suite: test_suite_formal.css file added. 1	
v0.14	\LWR@htmlsectionfilename: Fix: Links to home page. 190	v0.16	General: 2016/04/11 1
	General: 2016/03/31 1	\titlingpage: Improved print-output spacing. 244	
	floatrow: Added. 421	Added XeLaTeX, LuaLaTeX support. 118	
	Docs: Commands for a successful HTML conversion. 73	Docs: Font and UTF-8 support. . 66	
	Docs: Commands into a warpprint environment. 70	Docs: Moved location of \usepackage{lwarp}. 67	
	Docs: Newclude limitations. . . . 92	Docs: Text not converting. 101	
	Docs: Table: Cross-referencing data structures. 294	Fix: amsmath options clash 122	
		Fix: newtxmath compatibility. 122	
		Lwarp no longer selects fonts. 66, 118	
		Removed package: suffix 1	
		Test Suite: Improved titlingpage. 244	
		Test Suite: Lwarp no longer selects fonts. 1	

Test Suite: Supports XeLaTeX,	
LuaTeX.	1
xfrac: Adjusted for the use of any	
font.	543
v0.17	
\lwr@htmlsectionfilename: Fix:	
Links when entire doc is one	
HTML page.	190
General: 2016/04/14	1
mdframed: Added.	453
Test Suite: Fix: Print-version	
front-matter page numbers.	1
Test Suite: Mdframed	1
v0.18	
\lwr@hspace: \hspace supported. .	379
\lwr@includegraphicsb: Add: svgz	
file extension.	349
em, ex, %, px dimensions	
preserved.	349
Fix: \linewidth, \textwidth,	
\textheight inside a minipage. .	349
Improved HTML output linebreaks.	349
\lwr@myshorttoc: Reorganize	
\HomeHTMLfilename logic.	310
\lwr@newhtmlfile: sidetoc after	
title, improving responsive	
design.	220
\lwr@requesttoc: Reorganize	
\HomeHTMLfilename logic.	238
\lwr@subhyperref: Improved HTML	
output linebreaks.	301
\lwr@subhyperrefclass: Improved	
HTML output linebreaks.	301
\lwr@subinlineimage: Surpress	
extra space.	303
General: 2016/05/19	1
File: l warp.css: Improved TOC	
outline display.	1
Files: l warp.css and	
l warp_formal.css: Improved	
responsive design.	1
Microtype disabled during HTML	
generation	119
PDF Unicode input characters. .	107
Test Suite: Verse package	1
lateximage: pdfcrop: --hires added.	339
Reorganize \HomeHTMLfilename	
logic.	339
Surpress extra space.	339
verse: Supports verse, memoir	
packages.	531
minipage: Fix: \linewidth,	
\textwidth, \textheight	
inside a minipage.	364
v0.19	
\HTMLfilename: Docs: Escape	
filename underscores.	190
\HomeHTMLfilename: Docs: Escape	
filename underscores.	190
\lwr@l warpstart: Enabled \\ equal	
to \newline.	235
\lwr@doubledollar: MathJax	
support.	322
\lwr@filestart: l warp_mathjax.txt	
loaded.	232
\lwr@hspace: Fix: \hspace length	
computations.	379
\lwr@minipagestartpars:	
Surpresses paragraph tags	
between minipages.	378
\lwr@singledollar: MathJax	
support.	322
\lwr@tabledatamultirowtag:	
Added optional args.	467
\LateximageFontSizeName: Add:	
User-adjustable math/lateximage	
font size.	338
\minipagewidth: Added: No	
width tag for the next minipage	
in HTML.	364
\rule: Added	383
\warpHTMLonly: Added.	113
\warpprintonly: Replaces	
\rowprintedonly.	113
\xfracHTMLfontSize: Added.	543
General: 2016/06/08	1
Avoids MathJax.	321
cleveref: Loaded	
\AtEndPreamble.	360
CSS for table note item.	512
Docs: multirow browser bug	
workaround.	89, 260
Docs: Math options.	67
Docs: Table: Cross-referencing	
data structures, updated.	294
File: l warp_mathjax.txt added. .	1
File: l warp.css: tnoteitemheader	
added	1

Introduction: MathJax support	
mentioned.	33
MathJax support added.	323, 329, 330
Options: mathsvg and mathjax 111
Supports colored \rule. 538
titleps: null \pagestyle and	
\thispagestyle for HTML. . .	513
tikzpicture: Adapts to tikz version.	513
equation: MathJax support.	327
v0.20	
\BlockClassSingle: Renamed from	
"LWR@htmldivclassline".	203
\HTMLDescription: Added	
\NewHTMLdescription.	
(Renamed in v0.30)	212
\HTMLFilename: No longer escape	
underscores.	190
\HomeHTMLFilename: No longer	
escape underscores.	190
\InlineClass: Renamed from	
"inlineclass".	203
\LWR@LwarpStart: Fix: math cross	
references.	236
\LWR@closeparagraph: \unskip	
extra spaces.	206
No break tags in the start/end of a	
tabular.	206
\LWR@endofline: Fix: \\	377
\LWR@filestart: Adds meta	
description.	232
\LWR@hspace: Add: Supports HTML	
thin breakable space.	379
\LWR@htmldivclass: Added	
optional style.	201
\LWR@htmlelementclass: Added	
optional style.	201
\LWR@htmlsectionfilename:	
HTMLFilename: removed	
additional trailing '-', and may be	
empty.	190
Sections called "Index" or "index"	
have an underscore prepended to	
their filenames if no prefix. . .	190
\LWR@includegraphicsb: Fix:	
\linewidth in a floatrow.	349
Fix: Expands filename.	349
\LWR@longtabledatacaptiontag:	
Fix: Pars in captions.	283
\LWR@section: Combined	
higher-level sections together	
into files.	225
\LWR@setOSWindows: Auto-detects	
operating system.	109
\LWR@subhtmlelementclass:	
Factored code.	200
\SetHTMLFileName: Add: Control	
file numbers.	190
\cpagerefFor: User-redefinable	
word for page references.	361
\dotfill: Inserts an ellipsis.	376
\hfill: Inserts a \quad.	376
\hrulefill: Inserts a short rule. . .	376
\hyperindexref: Print mode	
provided in case hyperref not	
used.	318
\pageref: Added.	300
\tracingl warp: Added.	128
General: 2017/02/09	1
afterpage: Added.	390
alltt: Added.	391
bookmark: Added.	398
caption and subcaption supported. .	1
cleveref and referencing patches:	
Applied \AfterEndPreamble. .	360
draftwatermark: Added.	402
eso-pic: Added.	406
everypage: Added.	407
extramarks: Added.	408
fancyhdr: Added.	412
hyperref: Additional user macros. .	437
keyfloat: Added.	442
letterspace: User-interface	
emulated.	443
listings: Added.	445
ltcaption: Added.	451
lwarp-newproject: Added.	133
microtype: User-interface	
emulated.	462
needspace: Added.	469
nowidow: Added.	471
placeins: Added.	484
ragged2e: Added.	484
setspace: Improved support. . .	490
textpos: Added.	507
titleps: Added.	513
titlesec: Added.	515
titletoc: Added.	516
titling: Improved compatibility. .	517

tocloft: Added.	522
wallpaper: Added.	532
wrapfig: Added.	533
Added @, <, > columns.	260
Added single-expansion data arrays.	188
Code factored into independent \warp_html files.	387
Docs: Examples for generating HTML file names.	59
Docs: Improved index.	1
Enhanced titling support.	242
File: l warp.css: Minor fixes for validation.	1
File: l w arp mk used to compile print, HTML, indexes, and lateximages.	1
Fix: \ linewidth in a floatrow.	424
Improved float caption type handling.	419
Moved sidebar and example code to test suite.	1
Page geometry set to 6in wide with large margins.	120
Parallel versions of aux files for print/HTML.	1
Removed reliance on make, grep, gawk.	1
Tabular: \unskip extra spaces.	260
Test Suite: HTML meta descriptions.	1
Verbatim: Added.	413
verbatim: Added.	253
BlockClass: Added optional style.	202
Renamed from "blockclass".	202
LWR@nestspan: Fix: Minipages inside a span.	197
v0.21	
\ LWR@LwarpStart: Changed lateximages to a .txt file.	235
\ LWR@filestart: Skip title if not given.	232
\ LWR@newhtmlfile: Skip title if not given.	220
\ marginpar: Fixed source listing.	217
General: 2017/02/23	1
fontenc: Added.	428
fontspec: Added.	428
inputenc: Added.	441
newclude: Added.	469
v0.22	
newunicodechar: Added.	470
l w arp mk: Fix: l w arp mk again for Windows.	173
l w arp mk: Fix: l w arp mk l images for Windows.	173
l w arp mk: Fix: l w arp mk uses lateximages text file instead of shell script.	173
Add: Errors for misplaced packages.	114
Docs: Added internet class.	38
Docs: Added TeX2page, GladTeX.	38
Docs: Installing on Windows.	44
File l warp_tutorial.txt added.	48
v0.23	
\ LWR@parseDcolumn: Added tabular D column.	270
\ LWR@parsebangcolumn: Added tabular ! column.	268
\ LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns.	271
\ LWR@printmccoldata: Added tabular D, !, and X columns.	280
General: 2017/03/02	1
abstract: Added.	389
changepage: Added.	400
dcolumn: Added.	402
ftnright: Added.	433
geometry: Nullified commands.	434
indentfirst: Added.	441
layout: Added.	443
l escape: Added.	451
mcaption: Added.	453
nameref: Added.	469
nextpage: Added.	470
parskip: Added.	483
showkeys: Added.	491
sidecap: Added.	492
tabularx: Added.	504
varioref: Supported.	79
verse: Added.	531
v0.24	
\ LWR@parsetablecols: Fix for vert bar column type.	271
\ LWR@printmccoldata: Fix for vert bar column type.	280
General: 2017/03/02	1

v0.24	\LWR@hspace: Add: \hspace \fill converts to 2em	379
	\LWR@htmlfileref: Fix: Index links while \tracingl warp.	297
	\hypertocfloat: List of floats responds to lofdepth, lotdepth.	316
	General: 2017/03/15	1
	floatrow: Support for subfig.	421
	subfig: Added.	496
	tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary.	512
	Docs: Filename underscore.	57, 74
	No longer preloads subcaption; conflicted with subfig.	122
	picture: Fix for inline images.	361
	tikzpicture: Fix for inline images.	513
v0.25	\LWR@loadnever: Added the ability to prevent conflicting packages.	115
	\addcontentsline: Handles theorems.	309
	General: 2016/03/22	1
	amsthm: Added.	392
	ccaption: Prevented.	400
	ellipsis: Added.	403
	emptypage: Added.	403
	framed: Added.	431
	lips: Added.	445
	mdframed: Help avoid hyphenation.	454
	ntheorem: Added.	471
	showidx: Added.	491
	theorem: Added.	508
	Basic L ^A T _E X theorems: improved css.	253
	Docs: Adds credits for patched code.	1
	Docs: Testing l warp.	99
	Fix: Allows XE ^A L _E T _X and Lua ^A L _E T _X to preload graphics and graphicx.	115
v0.26	General: 2017/03/31	1
	cutwin: Added.	401
	endnotes: Added.	404
	floatflt: Added.	420
	footmisc: Added.	428
	footnotehyper: Added.	431
	footnote: Added.	429
	marginfix: Added.	452
	marginnote: Added.	452
	mparhack: Added.	465
	pagenote: Supported as-is.	482
	sidenotes: Added.	493
	l warp.css: Improved responsive marginpar and marginblock.	135
	Docs: Improved MiK _T E _X install instructions.	41, 44
	Dollar span avoided in a lateximage.	321
	Footnotes now are L ^A T _E X boxes instead of pagenotes.	213
	lateximage: Labels track page numbers of lateximages.	339
	Print mode now uses a minipage of \ linewidth.	339
	picture: Fix for \makebox in picture.	361
v0.27	\LWR@footnotetext: Fix for table footnote par tags.	214
	General: 2017/04/04	1
	lettrine: Added.	444
	microtype: Fix with Xe ^A L _E T _X , Lua ^A L _E T _X	462
	soul: Added.	495
	ulem: Added.	529
	Docs: Installing utilities for MacOS.	46
	Docs: Limitations of saveboxes.	75
	Page geometry modified to reduce line overflow.	120
v0.28	\@rindex: Improved indexing.	318
	\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.)	212
	\LWR@L warpEnd: If FormatEPUB or FormatWordProcessor, no bottom nav.	238
	\LWR@L warpStart: FormatWordProcessor forces single-file output.	235
	\LWR@filestart: Adds HTML meta author.	232
	\LWR@forcenewpage: Forces new PDF page before major environments.	194

\LWR@htmlcomment: Breaks ligatures in HTML comments.	200
\LWR@includegraphicsb: Adapts to graphics syntax.	349
\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav.	220
\LWR@parsetablecols: Added L, C, R, J column types.	271
\LWR@startref: Removed space. . .	299
\chapter: If EPUB, prints footnotes before each section.	230
\hyperindexref: Improved indexing.	318
\textup: Fixed span class.	373
General: 2017/04/14	1
glossaries: Added.	435
graphics: Added.	436
tabularx: Fix for optional pos. .	504
tabulary: Added.	504
\larpmk: Add: printglossary and htmlglossary commands. .	173
Added boolean FormatEPUB. . .	129
Added boolean FormatWordProcessor. . . .	129
Added boolean HTMLDebugComments. . . .	128
Added boolean HTMLMarkFloats	129
Docs: Modfyng \larpmk and index processing.	99
File \larp_mathjax.txt: Updated CDN repository. . .	171
Forced oneside to maintain large right margin.	120
v0.29	
\lWR@includegraphicsb: Fix: Error when no optional arguments. .	349
General: 2017/04/15	1
\larpmk: Add: language option for config files.	173
Add: \larpmklang option for \larp.	111
Docs: Using a glossary	64
File *.larpmkconf: Add: language option for config files. .	134
File \larpmk.conf: Add: language option for config files.	133
v0.30	
\CSSFilename: Renamed from \NewCSS.	211
v0.31	
General: 2017/05/15	1
keyfloat: Improved compatibility.	442
\HTMLAuthor: Renamed from \HTMLauthor.	212
\HTMLDescription: Renamed from \NewHTMLdescription.	212
\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	210
\HTMLLanguage: Renamed from \MetaLanguage.	232
\HTMLPageBottom: Renamed from \SetPageBottom.	211
\HTMLPageTop: Renamed from \SetPageTop.	211
General: 2017/04/29	1
\larp-newproject removed, and combined with \larp.	133
\larpmk: Add: xdyfile configuration option.	173
\larpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex.	173
\larpmk: Fix: xindy now used for print index generation with latexmk.	173
\larpmk: language now used for both index and glossary generation.	173
File: \larp_html.xdy renamed to \larp.xdy.	171
Fix: *.css files only written in print mode.	135
Fix: \larp.xdy only written in print mode.	171
Fix: \larp_mathjax.txt only written in print mode.	171
Option OSWindows replaces macro \warpOSWindows.	112
Option latexmk replaces macro \UseLatexmk.	112
Option \larpmklang changed to IndexLanguage.	111
Option xdyFilename added. . . .	111
Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename.	112

v0.32	
	\RequirePackage: Fix: Ignores blanks in package list. 124
	General: 2016/06/09 1
	glossaries: Prevent error with \glo@name not defined. 319
	lwarpmk: Fix: io.lines() changed to file.lines() due to luatex changes. 173
v0.33	
	\HTMLAuthor: Fix: Provides empty default author if none given. 212
	\LWR@loadbefore: Fix: No \PackageError if already loaded. 114
	\LWR@parseatcolumn: Fix: Column alignment with leftmost @. 267
	\LWR@tabledatasinglecolumntag: Fix: Macros in tabular could cause extra data cell. 276
	\LWR@vspace: Add: \vspace nullified. 380
	\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. 263
	General: 2017/07/10 1
	amsmath: Removed fleqn option. 122
	fancyhdr: Fix: Optional args for \lhead, etc. 412
	Add: Tabular at and bang columns now have their own HTML columns. 260
	cleveref: Fix: Loaded \AtEndPreamble. 360
	Fix: Incorrectly-inline math environments. 330
	New handling of & to localize catcode changes. 260
v0.34	
	\@fnsymbol: Text symbols instead of math. 245
	\InlineClass: Moved optional argument in front of mandatory. 203
	\LWR@htmldivclass: Moved optional argument in front of mandatory. 201
	\LWR@htmlelementclass: Moved optional argument in front of mandatory. 201
	\LWR@htmlelementclassline: Moved optional argument in front of mandatory. 201
	\LWR@htmlspanclass: Moved optional argument in front of mandatory. 198
	\LWR@nullfonts: Improved font control. 373
	\LWR@restoreorigformatting: booktabs: Works inside lateximage. 324
	Improved font control. 324
	\LWR@subhtmlelementclass: Moved optional argument in front of mandatory. 200
	\LWR@tabledatacolumntag: booktabs: Works inside lateximage. 289
	\fboxBlock: Added. 369
	\makebox: Fix: Handles paren arg. 367
	General: 2017/08/08 1
	babel-french: Adds fixed-width HTML spaces to punctuation. 196
	balance: Added. 396
	booktabs: Works inside lateximage. 291, 398
	boxedminipage2e: Added. 399
	boxedminipage: Prevented. 399
	crop: Added. 401
	enumerate: Added. 405
	enumitem: Added, no longer required. 405
	everyshi: Added. 407
	fancybox: Added. 409
	fancyvrb: Added, no longer required. 413
	figcaps: Added. 418
	filecontents: Required. Patched for morewrites. 121
	floatpag: Added. 421
	flushend: Added. 426
	fullpage: Added. 434
	hyperxmp: Added. 440
	idxlayout: Added. 440
	marginfit: Added. 451
	mdframed: Improved mdtheorem patch. 460
	moreverb: Added. 463
	paralist: Added. 482
	pdflscape: Added. 483

pdfsync: Added.	483
prelim2e: Added.	484
rotfloat: Added.	488
savetrees: Added.	489
shadow: Added.	491
syntonly: Added.	503
titlesp: No longer required.	513
titleref: Prevented.	515
xmpincl: Added.	545
Docs: Horizontal space limitations.	1
Docs: Misplaced alignment character.	101
File: README.txt: updated.	1
File: 1warp_mathjax.txt version change.	171
Fix: Added the eqnarray environments.	330
Improved font control.	372
Lists refactored to remove enumitem requirement.	255
Verbatim refactored to remove fancyvrb requirement.	251
lateximage: Fix: lateximage with minipage, \parbox, \makebox, \fbox, \framebox, \raisebox, \scalebox, \reflectbox.	339
fcolorminipage: Added.	541
BlockClass: Moved optional argument in front of mandatory.	202
fminipage: Added.	369
LWR@nestspan: Fix: Minipages, BlocksClass, and lists inside a span.	197
LWR@tabular: booktabs: Works inside lateximage.	291
v0.35	
General: 2017/08/08	1
Fix: \textbf and related.	372
v0.36	
\LWR@HTMLsanitize: Fix for babel-french.	338
\LWR@HTMLsanitizeexpand: Fix for babel-french.	339
\LWR@closeparagraph: Extra HTML source space after paragraphs.	206
\LWR@currenttextcolor: Fix for \textcolor when xcolor not loaded.	381
\LWR@footnotetext: Extra HTML source space after paragraphs.	214
Force HTML superscripts.	214
\LWR@nullfonts: Fix: Filenames while using MathJax.	373
\LWR@restoreorigformatting: siunitx: Improved super/subscripts in a lateXimage.	324
\LWR@section: Improved spacing.	225
\LWR@stopars: Extra HTML source space after paragraphs.	209
\fbox: Fix: Uses \fboxrule and \fboxsep.	368
\framebox: Fix: Handles width and horiz position.	368
\makebox: Fix: Handles width and horiz position.	367
General: 2017/08/17	1
babel-french: Adjustements for French variants, load order, footnotes, ellipses.	196
footnote: Extra HTML source space after paragraphs.	429
siunitx: Fix for babel-french.	344
siunitx: Improved symbol support.	494
transparent: Added.	527
upref: Added.	530
xcolor: Added \fcolorboxBlock, \colorboxBlock.	535
xcolor: Fix: Background none in print mode.	535
xcolor: Refactored \LWR@colorstyle.	538
xcolor: Uses \fboxrule and \fboxsep.	535
xcolor: \fcolorbox etc. now work inside lateximage.	535
Docs: Reorganized: Special cases and limitations.	73
Source: Improved formatting.	1
lateximage: Footnotes appear in regular text instead of the lateXimage minipage.	339
LWR@tabular: Fix for babel-french.	291
v0.37	
\include: Maintains independent .aux files for HTML.	126
General: 2017/08/19	1
\textcolor accents: Added.	131
babel-french: Adjustment for load order.	196

color: Prevented.	401	v0.39
comment: Maintains independent cutfiles for print, HTML.	113	
siunitx: Improved symbol support.	494	
textcomp: Improved support.	505	
lwarpmk: Removes additional HTML aux files.	173	
File handles reorganized.	125	
v0.38		
\@secntformat: Added for appendix.	225	
\ForceHTMLPage: Added.	223	
\ForceHTMLTOC: Added.	223	
\LWR@section: \part* starts a new HTML page, for appendix.	225	
Modified spacing, uses \numberline.	225	
\numberline: Added trailing \quad.	314	
\part: Fix with article class.	230	
General: 2017/08/27	1	
appendix: Added.	395	
arabicfront: Added.	395	
caption2: Prevented.	399	
chappg: Added.	400	
color: Forces xcolor as well.	401	
fix2col: Added.	418	
fncychap: Added.	427	
grffile: Added.	437	
metalogo: Added.	462	
nonumonpart: Added.	470	
nopageno: Added.	471	
pagenote: Option page disabled.	482	
realscripts: Added.	485	
relsize: Added.	486	
romanbarpagenumber: Added.	488	
romanbar: Added.	487	
scalefnt: Added.	489	
siunitx: Removed from lwarpmk core.	494	
textcomp: Removed from lwarpmk core.	505	
tocbibind: Added.	521	
xltxttra: Added.	545	
lwarpmk: Added print1 and html1 actions.	173	
Added \markboth, \sloppy, etc.	194	
Docs: Enhanced <i>Supported Functions</i> table.	34	
Docs: Index, tocbibind.	82	
Docs: Starred sections.	80	
\maketitle: titling version.	519	
Native L ^A T _E X version.	246	
Removed minipages.	246, 519	
Supports authblk with <div>s of class oneauthor instead of tabular.	246, 519	
\AddSubtitlePublished: Added.	248	
\LWR@domulticolumn: Add: Optional vpos and # rows.	281	
\LWR@restoreorigformatting: Appended with \appto instead of calling various macros.	324	
\LWR@tabledataacolumntag: Don't start a data cell if see \TabularMacro.	289	
\LWR@tabledatamultirowtag: Add: New optional vpos argument.	467	
Add: Supports left/right border for bigdelim.	467	
Fix: Long text argument.	467	
\ResumeTabular: Added.	288	
\TabularMacro: Added.	288	
\multicolumnrow: Added.	287, 468	
\printauthor: Removed minipages.	243	
Supports authblk with <div>s of class oneauthor instead of tabular.	243	
\thanksmarkseries: Removed minipage footnotes.	521	
General: 2017/09/05	1	
a4wide: Added.	388	
a4: Added.	388	
a5comb: Added.	388	
addlines: Added.	390	
anysize: Added.	394	
authblk: Added.	395	
bigdelim: Added.	397	
bigstrut: Added.	397	
chngpage: Prevented.	400	
ebook: Added.	403	
fullwidth: Added.	434	
supertabular: Added.	502	
textarea: Added.	505	
titling: Improved compatibility.	517	
titling: Removed extraneous center environments.	518	
typearea: Added.	528	
xtabular: Added.	546	
zwpagelayout: Added.	547	

Docs: Reorganized tabular discussion.	88	titlepage: Clear pending footnotes. 242
Titlepage \published and \subtitle removed.		Removed minipages. 242
\AddSubtitlePublished restores.	247	titlingpage: Clear pending footnotes. 517

§ 233 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	~	73
\\$	<i>321</i>	
\&	<i>189</i> , <u>4797</u>	
\(.	<u>5944</u>	A
\)	<u>5944</u>	a4 (package) <u>388</u>
\,	<u>73</u>	a4wide (package) <u>388</u>
\@osetpageref	<u>6908</u>	a5comb (package) <u>388</u>
\@osetcref	<u>6904</u>	abstract (environment) <u>4551</u>
\@osetcrefrange	<u>6905</u>	abstract (package) <u>388</u>
\@author	<u>240</u>	\abstractname <u>59</u> , <u>4550</u>
\@begintheorem	<u>4643</u>	accents <u>222</u>
\@capttype	<u>5633</u>	\addcontentsline <u>5687</u>
\@currentlabelname	<u>5413</u>	addlines (package) <u>390</u>
\@date	<u>240</u>	\AddSubtitlePublished <u>4513</u>
\@dblfloor	<u>5611</u>	Adobe (program) <u>40</u>
\@endtheorem	<u>4653</u>	\affiliation <u>4375</u>
\@ensuredmath	<u>5951</u>	afterpage (package) <u>120</u>
\@float	<u>5611</u>	afterpage (package) <u>390</u>
\@fnsymbol	<u>4465</u>	algorithmic
\@footnotetext	<u>3795</u>	with newfloat, trivfloat <u>528</u>
\@include	<u>440</u>	algorithmicx (package) <u>391</u>
\@makecaption	<u>5635</u>	align (environment) <u>6265</u>
\@makefnmark	<u>3767</u>	align* (environment) <u>6288</u>
\@makefntext	<u>3766</u>	alignment tab character &, misplaced <u>88</u> , <u>258</u> , <u>263</u>
\@maketitle	<u>49</u> , <u>4488</u>	alltt (package) <u>391</u>
\@mpfootnotetext	<u>3796</u>	\AmS <u>7418</u>
\@opargbegintheorem	<u>4648</u>	AMSmath
\@secCntformat	<u>4024</u>	split miss-numbered <u>83</u> , <u>85</u> , <u>319</u> , <u>472</u>
\@title	<u>240</u>	amsmath (package) <u>122</u>
\@wrglossary	<u>5885</u>	amsthm (package) <u>392</u>
\@wrindex	<u>5878</u>	\and <u>240</u>
\\"	<u>377</u>	any size (package) <u>394</u>
\\$	<u>322</u>	appendix (package) <u>395</u>
\\$\\$	<u>322</u>	arabicfront (package) <u>395</u>
		array (package) <u>294</u>

AsciiDoc (program)	39	LWR@validtablecol	271
AsciiDoctor (program)	39	LWR@verbtags	252
Asciidoctor-LaTeX (program)	39	mathjax	110
\attrib	92, 250, 531	usingOSWindows	109
\attribution	4561	warpingHTML	110
authblk (package)	395	warpingprint	110
author		boxedminipage (package)	399
HTML meta tag	71, 212	boxedminipage2e (package)	399
\author	71, 240	\boxframe	137
autosec	224	bugs	101
		BVerbatim (environment)	183
B			
babel (package)	93	C	
\backmatter	4008	calc (package)	122
balance (package)	396	Calibre	94
\BaseJobname	3216	caption (package)	122
BaseJobname (option)	111	caption2 (package)	399
baseline		\caption@begin	5666
tabular	271	\caption@end	5666
\bfseries	7222	\captionlistentry	5670
\BibTeX	7413	\captionof	5700
bigdelim (package)	397	\CaptionSeparator	5634
bigstrut (package)	397	ccaption (package)	400
BlockClass (environment)	3541	center (environment)	6500
\BlockClassSingle	3553	\centering	236
bookmark (package)	398	changepage (package)	400
booktabs (package)	398	chappg (package)	400
booleans:		\chapter	4167
CombineHigherDepths	58, 217	chngpage (package)	400
FileSectionNames	59, 190	\citetitle	4568
FormatEPUB	94, 129	classss:	
FormatWordProcessor	96, 129	internet	38
HTMLDebugComments	128	cleveref (package)	360
HTMLMarkFloats	97, 129	cmap (package)	67
LWR@camsmultiline	329	\color	538
LWR@doingapar	205	color (package)	401
LWR@doinghline	261	\colorbox	45
LWR@doingstartpars	204	\colorboxBlock	55, 6800
LWR@doingtbrule	261	CombineHigherDepths (boolean)	58, 217
LWR@freezethisfloat	306	comment (package)	113
LWR@infloatarrow	349	Computer Modern	66
LWR@intabularmetadata	262	\ConTeXt	7410
LWR@minipagewidth	364	counters:	
LWR@minipagethispar	364	FileDepth	58, 217
LWR@skipatbang	262	lofdepth	315
LWR@skippingmrowcell	262	lotdepth	315
LWR@starredlongtable	283	LWR@externalfilecnt	321
LWR@startedrow	261	LWR@htmlfilenumber	190
LWR@tableparcell	262	LWR@lateautopage	307
LWR@tracingl warp	128	LWR@lateximagedepth	337

LWR@lateXimageNumber	337	\end@float	5625
LWR@LIPage	338	\EndDefiningTabulars	4791
LWR@midoRuleCounter	277	endnotes (package)	404
LWR@minipagedepth	362	\enlargeThisPage	7341
LWR@nextautofloat	307	\enskip	378, 7302
LWR@nextautopage	307	enumerate (environment)	4679
LWR@nextEquation	323	enumerate (package)	404
LWR@prevFileDepth	225	enumitem (package)	405
LWR@spandepth	204	environ (package)	122
LWR@tablecolspos	266	environments:	
LWR@tablecolswidth	266	abstract	4551
LWR@tabletotalcols	266	align	6265
LWR@tabletotalcolsnext	266	align*	6288
LWR@thisfloat	306	BlockClass	3541
SideTOCDepth	58, 313	BVerbatim	183
tocdepth	57	center	6500
\cpagerefFor	6907	description	4707
crop (package)	401	displaymath	5962
cross-references		enumerate	4679
missing or incorrect	101	eqnarray	6121
CSS		equation	6072
file selection	72	fcolorminipage	113, 6842
lwarf.css	72	flalign	6311
per HTML page	72	flalign*	6334
project-specific changes	72	flushleft	6512
\CSSfilename	56, 59, 72, 3737	flushright	6506
cutwin (package)	401	fminipage	76, 7094, 7142
D		gather	6219
danger icon	106	gather*	6242
\date	71	itemize	4665
dcolumn (package)	402	latextimage	6410
debugging	101	longtable	2
HTML debug comments	128	LWR@nestspan	3389
\DeclareGraphicsExtensions	6538	LWR@tabular	5351
\DeclareGraphicsRule	6538	LWRprint@fminipage	7118
Deja Vu	66	math	5961
description		minipage	6970
HTML meta tag	71, 212	multiline	6170
description (environment)	4707	multiline*	6194
displaymath (environment)	5962	picture	361, 6925
\dotfill	7238	quote	4576
draftwatermark (package)	402	theindex	5856
E		tikzpicture	8
ebook (package)	403	titlepage	70, 4392
ellipsis (package)	403	titlingpage	14, 70
\emph	7146	Verbatim	13
emptypage (package)	403	verbatim	4633
\end@dblfloat	5625	verse	2
		warpall	69, 100
		warpHTML	50, 61, 69, 101

<i>warpprint</i>	60, 69, 101	<i>tutorial.tex</i>	48
<i>epigraph</i> (package)	405	<i>FileSectionNames</i> (boolean)	59, 190
EPUB		<i>fix2col</i> (package)	418
conversion software	94	<i>\flagverse</i>	532
HTML conversion settings	94, 129	<i>\flalign</i> (environment)	6311
<i>eqnarray</i> (environment)	6121	<i>\flalign*</i> (environment)	6334
equation		<i>Flare</i> (program)	40
miss-numbered	83, 85, 319, 472	<i>float</i> (package)	419
<i>equation</i> (environment)	6072	<i>floatflt</i> (package)	420
error messages	101	<i>\floatname</i>	9
<i>eso-pic</i> (package)	406	<i>floatpag</i> (package)	421
<i>etoolbox</i> (package)	108	<i>\floatplacement</i>	12
<i>everyhook</i> (package)	121	<i>floatrow</i> (package)	421
<i>everypage</i> (package)	407	<i>\floatstyle</i>	15
<i>everyshi</i> (package)	407	<i>\flushbottom</i>	3310
<i>exp13</i> (package)	120	<i>flushend</i> (package)	426
<i>extramarks</i> (package)	408	<i>flushleft</i> (environment)	6512
		<i>flushright</i> (environment)	6506
		<i>fminipage</i> (environment)	76, 7094, 7142
		<i>fncychap</i> (package)	427
		<i>font</i>	
<i>fancybox</i> (package)	408	bitmapped	66
<i>fancyhdr</i> (package)	412	Computer Modern	66
<i>fancyvrb</i> (package)	413	Deja Vu	66
FAQ	101	ligatures	66, 67
fbox		packages	66
around a minipage	76	selection	66
<i>\fbox</i>	76, 7077	size — lateximage	84, 319, 337
<i>\fboxBlock</i>	76, 7086, 7141	size — math, SVG	84, 319, 337
<i>\fcolorbox</i>	67, 6806	size — xfrac	543
<i>\fcolorboxBlock</i>	87, 6841	<i>fontenc</i> (package)	66, 428
<i>fcolorminipage</i> (environment)	113, 6842	<i>fontspec</i> (package)	66, 428
<i>figcaps</i> (package)	418	<i>footmisc</i> (package)	428
file		<i>footnote</i> (package)	429
inaccessible	58	<i>footnotehyper</i> (package)	430
underscore	57, 74	<i>footnotes</i>	213
<i>filecontents</i> (package)	121	for	
<i>FileDepth</i> (counter)	58, 217	HTML & PRINT	106
files:		HTML output	106
<i>glyptounicode</i>	67	PRINT output	106
<i>lwarf.css</i>	72, 135	<i>\ForceHTMLPage</i>	80, 3983
<i>lwarf.xdy</i>	72, 171	<i>\ForceHTMLTOC</i>	80, 3989
<i>lwarf_formal.css</i>	166	<i>FormatEPUB</i> (boolean)	94, 129
<i>lwarf_mathjax.txt</i>	171	<i>FormatWordProcessor</i> (boolean)	96, 129
<i>lwarf_sagebrush.css</i>	161	<i>\framebox</i>	7056
<i>lwarf_tutorial.txt</i>	48	<i>framed</i> (package)	431
<i>lwarpmk.conf</i>	133	framed objects	76
<i>lwarpmk.lua</i>	99	<i>FrameMaker</i> (program)	40
<i>project.css</i>	72	Frequently Asked Questions	101
<i>project.lwarpmkconf</i>	134	<i>\frontmatter</i>	4005
<i>project_html.tex</i>	133		
<i>sample_project.css</i>	72, 170		

ftnright (package)	433
fullpage (package)	434
fullwidth (package)	434
\fup	7235
\fussy	<u>3312</u>
G	
gather (environment)	6219
gather* (environment)	6242
GELLMU (program)	38
generator	
HTML meta tag	233
geometry (package)	120, 434
gettitlestring (package)	120
GladTeX (program)	39
glossaries (package)	82, 435
glossary	
language	64
processing	64
glyptounicode (file)	67
graphics (package)	345, 436
\graphicspath	<u>6536</u>
graphicx (package)	345, 436
grffile (package)	437
H	
Hevea (program)	38
\hfill	7236
\HomeHTMLFilename	3218
HomeHTMLFilename (option)	57, 112
horizontal space	73
between minipages	378
\href	5556
\hrulefill	<u>7237</u>
\hskip	73
\hspace	73, 378, <u>7332</u>
HTML	
commands for a successful conver-	
sion	73
conversion settings	57
debug comments	128
EPUB	94, 129
word processor	96, 97, 129
filename generation	59
headings	104
in place of SVG images	54, 102
meta tag	
author	71, 212
description	71, 212
generator	233
viewport	233
tabular column conversion	272
\HTMLAuthor	60, 71, <u>3751</u>
HTMLDebugComments (boolean)	128
\HTMLDescription	60, 71, <u>3757</u>
\HTMLEntity	<u>3199</u>
\HTMLFilename	<u>3217</u>
HTMLFilename (option)	57, 112
\HTMLFirstPageTop	60, 3714
htmlglossary (option) [lwarpmk]	82, 435
\HTMLLanguage	60, <u>4223</u>
HTMLleftmargini (length)	92, 250, 251, <u>531</u>
HTMLMarkFloats (boolean)	97, 129
\HTMLPageBottom	60, 240, <u>3730</u>
\HTMLPageTop	60, <u>3722</u>
\HTMLUnicode	<u>3207</u>
HTMLvleftskip (length)	92, 250, 251, <u>531</u>
\hyperindexref	<u>5892</u>
hyperref (package)	301, 437
\hypertoc	<u>5802</u>
\hypertocfloat	<u>5815</u>
hyperxmp (package)	<u>440</u>
I	
icon	
warning	106
idxlayout (package)	<u>440</u>
ifplatform (package)	<u>108</u>
images	
graphicx package	345
in strange places	102
showing as HTML	54, 102
\include	<u>440</u>
\includegraphics	<u>6728</u>
indentfirst (package)	<u>441</u>
InDesign (program)	<u>40</u>
index	
language	64
placement and TOC options	82
processing	52, 53
UTF-8	67
IndexLanguage (option)	64, 82, 111, <u>435</u>
\InlineClass	<u>3561</u>
inputenc (package)	66, <u>441</u>
internet (class)	<u>38</u>
item	
empty	74, 255
itemize (environment)	<u>4665</u>
\itshape	<u>7227</u>

	J	
JavaScript		
MathJax	84, 319
	K	
\kern	73
keyfloat (package)	442
kvoptions (package)	110
	L	
label		
in HTML	236
math environment	329
language		
glossary	64
index	64
language HTML metadata	232
\LaTeX	7396
\TeX2HTML (program)	38
\LaTeXe	7396
lateximage (environment)	6410
\LateximageFontSizeName	6368
lateximages		
font size	84, 319, 337
latexmk (option)	57, 112
LaTeXML (program)	38
layout (package)	443
lengths:		
HTMLleftmargini	...	92, 250, 251, 531
HTMLvleftskip	...	92, 250, 251, 531
LWR@minipageheight	363
LWR@minipagewidth	362
LWR@tempheight	379
LWR@tempraise	379
LWR@tempwidth	379
\VerbatimHTMLWidth	251
vleftmargini	92, 250, 531
vleftskip	92, 250, 531, 532
letltxmacro (package)	108
letterspace (package)	443
lettrine (package)	444
LibreOffice (program)	39
ligatures	66, 67, 119
line numbers	106
\linebreak	7335
\LinkHome	3268
Linux (program)	68, 109
lips (package)	445
list item, empty	74, 255
listings (package)	445
	\	
listof	5768
modern (package)	66
lofdepth (counter)	315
longtable (environment)	2
longtable (package)	449
lotdepth (counter)	315
lscape (package)	451
ltcaption (package)	451
	L	
LuaLaTeX		
detection	118
file & section names	222
\LuaLaTeX	7404
LuaLaTeX (program) [requirement]	42
\LuaTeX	7404
l warp		
loading	67
options	67
l warp (package)	67
l warp.css (file)	72, 135
l warp.xdy		
customizing	72
l warp.xdy (file)	72, 171
l warp_formal.css (file)	166
l warp_mathjax.txt (file)	171
l warp_sagebrush.css (file)	161
l warp_tutorial.txt (file)	48
l warpmk		
customizing	99
l warpmk:		
htmlglossary (option)	82, 435
printglossary (option)	82, 435
l warpmk (option)	111
l warpmk (program)	99, 173
l warpmk.conf (file)	133
l warpmk.lua (file)	99
l LWR@addmathjax	6061
l LWR@afterendverbatim	4617
l LWR@amsmultiline (boolean)	329
l LWR@atbeginverbatim	4603
l LWR@beginhideamsmath	6106
l LWR@blackborderpadding	7072
l LWR@borderpadding	28
l LWR@botnavigation	3276
l LWR@caption@begin	5647
l LWR@caption@end	5657
l LWR@clearmidrules	5051
l LWR@closeparagraph	3629
l LWR@closeprevious	3296
l LWR@closetabledatcell	4756
l LWR@colaferspec	267

LWR@colatspec	267	\LWR@htmlmathlabel	6092
LWR@colbangspec	267	\LWR@htmlemulticolumn	5167
LWR@colbeforespec	267	\LWR@htmlopencomment	3455
\LWR@copyfile	475	\LWR@htmlrefsectionfilename	3261
\LWR@createautosec	4014	\LWR@HTMLsanitize	6371
\LWR@currentcss	3736	\LWR@HTMLsanitizeexpand	6392
\LWR@currenttextcolor	7342	\LWR@htmlsectionfilename	3228
\LWR@descitem	4693	\LWR@htmlspan	3430
\LWR@docmidrule	5070	\LWR@htmlspanclass	3438
LWR@doingapar (boolean)	205	\LWR@htmntag	3446
LWR@doinghline (boolean)	261	\LWR@htmntagc	3379
LWR@doingstartpars (boolean)	204	\LWR@includographicsb	6630
LWR@doingtbrule (boolean)	261	\LWR@indexitem	5866, 5870, 5874
\LWR@domulticolumn	5144	\LWR@infloatarrow (boolean)	349
\LWR@doubledollar	5918	\LWR@intabularmetadata (boolean)	262
\LWR@endhideamsmath	6114	\LWR@itemizeitem	4658
\LWR@endminipage	6934	\LWR@latestautopage (counter)	307
\LWR@endofline	7285	\LWR@lateximagedepth (counter)	337
\LWR@endsubminipage	6955	\LWR@lateximagedepthref	5443
\LWR@ensuredoingapar	3602	\LWR@lateximagenumber (counter)	337
\LWR@epubprintpendingfootnotes	3833	\LWR@lateximagenumberref	5446
LWR@externalfilecnt (counter)	321	\LWR@lateximagesfile	437
\LWR@filenamenoblocks	3853	\LWR@LIPage (counter)	338
\LWR@filestart	4230	\LWR@loadafter	111
\LWR@findword	369	\LWR@loadbefore	122
\LWR@floatbegin	5595	\LWR@loadnever	131
\LWR@floatend	5613	\LWR@longtabledatacaptiontag	5178
\LWR@footnotetext	3768	\LWR@lookforpackagename	372
\LWR@forceminwidth	7061	\LWR@LwarpEnd	4352, 7440
\LWR@forcenewpage	3302	\LWR@LwarpStart	4280, 7440
LWR@freezethisfloat (boolean)	306	\LWR@maketitlesetup	33, 4457
\LWR@futurenonospacelet	4733	\LWR@maybeincthisfloat	5630
\LWR@FVstyle	17	\LWR@maybenewtablerow	4981
\LWR@getexparray	3195	\LWR@midrulecounter (counter)	277
\LWR@getmynexttoken	4740	\LWR@midrules	277
\LWR@hidelatexequation	6047	\LWR@minipagedepth (counter)	362
\LWR@hspace	7310	\LWR@minipagefullwidth (boolean)	364
\LWR@htmlblockcomment	3483	\LWR@minipageheight (length)	363
\LWR@htmlblocktag	3491	\LWR@minipagestartpars	7290
\LWR@htmleclosecomment	3455	\LWR@minipagestoppars	7293
\LWR@htmlcomment	3476	\LWR@minipagethispar (boolean)	364
\LWR@htmldivclass	3518	\LWR@minipagewidth (length)	362
\LWR@htmldivclassend	3521	\LWR@multicolother	5095
\LWR@htmlelement	3531	\LWR@multicolpartext	5090
\LWR@htmlelementclass	3505	\LWR@multicolskip	5101
\LWR@htmlelementclassend	3510	\LWR@multirowborder	3
\LWR@htmlelementclassline	3524	\LWR@mynexttoken	4732
\LWR@htmlelementend	3534	\LWR@myshorttoc	5712
LWR@htmlfilenumber (counter)	190	\LWR@nameref	5436
\LWR@htmlfileref	5439	\LWR@nestspan (environment)	3389

\LWR@newhtmlfile	3919	\LWR@spandepth (counter)	204
\LWR@newlabel	5473	\LWR@splabel	5449
\LWR@newref	5523	\LWR@starredlongtable (boolean)	283
\LWR@nextautofloat (counter)	307	\LWR@startedrow (boolean)	261
\LWR@nextautopage (counter)	307	\LWR@startnewdepth	4018
\LWR@nextequation (counter)	323	\LWR@startpars	3666
\LWR@nohspace	7331	\LWR@startref	5486
\LWR@nullfonts	7190	\LWR@stoppars	3682
\LWR@openparagraph	3607	\LWR@stripperiod	5414
\LWR@origcolsing	4827	\LWR@strresult	4825
\LWR@originname	6740	\LWR@subcmidrule	5062
\LWR@originnames	6747	\LWR@subhtmelementclass	3496
\LWR@parseaftercolumn	4887	\LWR@subhyperref	5547
\LWR@parseatcolumn	4833	\LWR@subhyperrefclass	5551
\LWR@parsebangcolumn	4857	\LWR@subinlineimage	5588
\LWR@parsebeforecolumn	4878	\LWR@subminipage	6949
\LWR@parseDcolumn	4915	\LWR@subnewref	5517
\LWR@parseenormalcolumn	4900	\LWR@subsublabel	5452
\LWR@parsepcolumn	4911	\LWR@subtableofcontents	5727
\LWR@parseskipcolumn	4897	\LWR@syncmathjax	5964
\LWR@parsetablecols	4920	\LWR@tablecolsing	4824
\LWR@patchlists	4721	\LWR@tablecolspos (counter)	266
\LWR@prevFileDepth (counter)	225	\LWR@tablecolswidth (counter)	266
\LWR@printatbang	4997	\LWR@tabledatacolumnntag	5289
\LWR@printmcoldata	5104	\LWR@tabledatamultirowtag	4
\LWR@printmcoltpe	5074	\LWR@tabledatasinglecolumntag	5011
\LWR@printpendingfootnotes	3823	\LWR@tableparcell (boolean)	262
\LWR@printthetitle	4416	\LWR@tabletotalcols (counter)	266
\LWR@ProvidesPackageDrop	424	\LWR@tabletotalcolsnext (counter)	266
\LWR@ProvidesPackagePass	412	\LWR@tabular (environment)	5351
\LWR@pushoneclose	4017	\LWR@tabularendofline	4808
\LWR@quickfile	434	\LWR@tabularhtmlcolumns	5244
\LWR@requesttoc	4345	\LWR@tempheight (length)	379
\LWR@requirepackagenames	368	\LWR@tempraise (length)	379
\LWR@restoreorigaccents	608	\LWR@tempwidth (length)	379
\LWR@restoreorigformatting	5984	\LWR@thisfilename	3851
\LWR@restoreorigprintxcolor	6880	\LWR@thisfloat (counter)	306
\LWR@rotboxorigin	6739	\LWR@thisnewfilename	3852
\LWR@rotstyle	6619	\LWR@titlingmaketitle	76, 4505
\LWR@rule	7343	\LWR@topnavigation	3273
\LWR@scalestyle	6622	\LWR@traceinfo	496
\LWR@section	4025	\LWR@tracingl warp (boolean)	128
\LWR@sectionnumber	4011	\LWR@validtablecol (boolean)	271
\LWR@setexpparray	3190	\LWR@verbtags (boolean)	252
\LWR@setlatestname	5415	\LWR@vspace	7333
\LWR@setOSWindows	29	\LWRprint@fminipage (environment)	7118
\LWR@sidetoc	5781	\LWRsetnextfloat	5638
\LWR@singledollar	5930	\LyX	7421
\LWR@skipatbang (boolean)	262		
\LWR@skippingmrowcell (boolean)	262		

M	
Mac OS (program)	<i>68, 109</i>
Madcap (program)	<i>40</i>
\mainmatter	<i>4001</i>
\makebox	<i>7033</i>
makeidx (package)	<i>121</i>
\MakeIndex	<i>7413</i>
\maketitle	<i>40, 70, 4469</i>
margin	
numbers	<i>106</i>
tags	<i>106</i>
marginfit (package)	<i>451</i>
marginfix (package)	<i>452</i>
marginnote (package)	<i>452</i>
\marginpar	<i>3840</i>
\markboth	<i>3307</i>
\markright	<i>3308</i>
markup languages	<i>40</i>
math	
font size — SVG	<i>84, 319, 337</i>
mathjax option	<i>111</i>
MathJax summary	<i>84, 320</i>
mathsvg option	<i>111</i>
showing as HTML	<i>54, 102</i>
SVG summary	<i>84, 319</i>
math (environment)	<i>5961</i>
MathJax	
mathjax option	<i>111</i>
subequations	<i>85, 320</i>
summary	<i>84, 320</i>
tagged equations	<i>85, 320</i>
MathJax (program)	<i>84, 85, 320</i>
MathJax (program) [requirement]	<i>42</i>
mathjax (boolean)	<i>110</i>
mathjax (option)	<i>57, 67, 111</i>
mathsvg (option)	<i>57, 67, 111</i>
\mcaption (package)	<i>453</i>
\mdframed (package)	<i>453</i>
\mdseries	<i>7221</i>
memoir	
verse	<i>532</i>
memoir (package)	<i>532</i>
meta tag, HTML	
author	<i>71, 212</i>
description	<i>71, 212</i>
generator	<i>233</i>
viewport	<i>233</i>
metalogo (package)	<i>462</i>
microtype (package)	<i>67, 119, 462</i>
\MiKTeX	<i>7420</i>
minipage	
framed	<i>76</i>
horizontal space between	<i>378</i>
minipage (environment)	<i>6970</i>
\minipagewidth	<i>6962</i>
misplaced \noalign	<i>89, 260</i>
misplaced alignment tab character &	<i>88, 258, 263</i>
missing sections	<i>58</i>
moreverb (package)	<i>463</i>
\mparhack (package)	<i>465</i>
\mrowcell	<i>5340</i>
MS-Windows (program)	<i>68, 109</i>
multicol (package)	<i>465</i>
multicolumn	
with multirow	<i>468</i>
\multicolumnrow	<i>26, 5265</i>
multirow	
with multicolumn	<i>468</i>
multirow (package)	<i>466</i>
multiline (environment)	<i>6170</i>
multiline* (environment)	<i>6194</i>
N	
\Nameref	<i>5544</i>
\nameref	<i>5535</i>
nameref (package)	<i>469</i>
needspace (package)	<i>469</i>
newclude (package)	<i>92, 469</i>
newfloat	
with trivfloat, algorithmic	<i>528</i>
\newfloat	<i>2, 4</i>
newfloat (package)	<i>122</i>
\newline	<i>7283</i>
\newtheorem	<i>253</i>
newtxmath (package)	<i>86</i>
newunicodechar (package)	<i>66, 470</i>
nextpage (package)	<i>470</i>
nicefrac (package)	<i>86</i>
\nolinebreak	<i>7336</i>
\nolinkurl	<i>5568</i>
nonumonpart (package)	<i>470</i>
\nopagebreak	<i>7340</i>
\nopagecolor	<i>44</i>
nopageno (package)	<i>471</i>
\normalfont	<i>7229</i>
nowidow (package)	<i>471</i>
ntheorem (package)	<i>471</i>
\numberline	<i>5799</i>

numbers	
left margin	106
numindex (option) [tocbibind]	83
O	
OpenOffice (program)	39
options:	
[lwarpmk]:	
htmlglossary	82, 435
printglossary	82, 435
[tocbibind]:	
numindex	83
BaseJobname	111
HomeHTMLFilename	57, 112
HTMLFilename	57, 112
IndexLanguage	64, 82, 111, 435
latexmk	57, 112
lwarpmk	111
mathjax	57, 67, 111
mathsvg	57, 67, 111
OSWindows	68, 109, 112
warpHTML	67, 110
warpprint	67, 110
xdyFilename	73, 111
\OSPathSymbol	28
OSWindows (option)	68, 109, 112
P	
packages	
required	118
packages:	
a4	388
a4wide	388
a5comb	388
abstract	388
addlines	390
afterpackage	120
afterpage	390
algorithmicx	391
alltt	391
amsmath	122
amsthm	392
any size	394
appendix	395
arabicfront	395
array	294
authblk	395
babel	93
balance	396
bigdelim	397
bigstrut	397
bookmark	398
booktabs	398
boxedminipage	399
boxedminipage2e	399
calc	122
caption	122
caption2	399
ccaption	400
changepage	400
chappg	400
chngpage	400
cleveref	360
cmap	67
color	401
comment	113
crop	401
cutwin	401
dcolumn	402
draftwatermark	402
ebook	403
ellipsis	403
emptypage	403
endnotes	404
enumerate	404
enumitem	405
environ	122
epigraph	405
eso-pic	406
etoolbox	108
everyhook	121
everypage	407
everyshi	407
expl3	120
extramarks	408
fancybox	408
fancyhdr	412
fancyvrb	413
figcaps	418
filecontents	121
fix2col	418
float	419
floatflt	420
floatpag	421
floatrow	421
flushend	426
fncychap	427
fontenc	66, 428
fontspec	66, 428
footmisc	428

footnote	429	nextpage	470
footnotehyper	430	nicefrac	86
framed	431	nonumonpart	470
ftnright	433	nopageno	471
fullpage	434	nowidow	471
fullwidth	434	ntheorem	471
geometry	120, 434	pagenote	482
getttitlestring	120	paralist	482
glossaries	82, 435	parskip	483
graphics	345, 436	pdflscape	483
graphicx	345, 436	pdfsync	483
grffile	437	placeins	484
hyperref	301, 437	prelim2e	484
hyperxmp	440	ragged2e	484
idxlayout	440	realscripts	485
ifplatform	108	refcount	122
indentfirst	441	relsize	486
inputenc	66, 441	romanbar	487
keyfloat	442	romanbarpagenumber	488
kvoptions	110	rotating	488
layout	443	rotfloat	488
letltxmacro	108	savetrees	489
letterspace	443	scalefnt	489
letrine	444	setspace	490
lips	445	shadow	491
listings	445	showidx	491
lmodern	66	showkeys	491
longtable	449	sidecap	492
lscape	451	sidenotes	493
ltcaption	451	siunitx	85, 344, 494
lwarp	67	soul	495
makeidx	121	subfig	496
marginfit	451	supertabular	502
marginfix	452	syntonly	503
marginnote	452	tabularx	504
mcaption	453	tabulary	504
mdframed	453	textarea	505
memoir	532	textcomp	67, 505
metalogo	462	textpos	507
microtype	67, 119, 462	theorem	508
moreverb	463	threeparttable	511
mparhack	465	tikz	512
multicol	465	titleps	513
multirow	466	titleref	515
nameref	469	titlesec	515
needspace	469	titletoc	516
newclude	92, 469	titling	517
newfloat	122	tocbibind	83, 521
newtxmath	86	tocloft	522
newunicodechar	66, 470	transparent	527

trivfloat	527	prelim2e (package)	484
typearea	528	\printauthor	240, 4424, 4443
ulem	529	\printdate	240, 4435, 4445
units	86	printglossary (option) [lwarpmk]	82, 435
upref	530	\printindex	5849
verse	531, 532	\PrintStack	3281
wallpaper	532	\printthanks	240, 242
wrapfig	533	\printtitle	240, 4408, 4442
xcolor	534	problems	101
xfrac	123, 542	programs	
xifthen	121	utility	41
xtlxtra	545	programs:	
xmpincl	545	[requirement]:	
xparse	120	LuaTeX	42
xstring	121	MathJax	42
xtab	546	pdfcrop	42
zref	122	pdfLaTeX	42
zwpagelayout	547	pdfseparate	42, 46
page		pdfcairo	42, 46
inaccessible	58	pdftotext	42, 46
\pagebreak	7337	perl	47
\pagecolor	43	XeLaTeX	42
\pagernote (package)	482	Adobe	40
\pagenumbers	3313	AsciiDoc	39
\pageref	5530	AsciiDoctor	39
\pagerefPageFor	5529	Asciidoc-LaTeX	39
\pagestyle	3305	Flare	40
Pandoc (program)	39	FrameMaker	40
\paragraph	4205	GELLMU	38
paralist (package)	482	GladTeX	39
\parbox	7026	Hevea	38
\parsemulticolumnalignment	5131	InDesign	40
parskip (package)	483	LaTeX2HTML	38
\part	4159	LaTeXML	38
pdfcrop (program) [requirement]	42	LibreOffice	39
pdfLaTeX (program) [requirement]	42	Linux	68, 109
pdfLandscape (package)	483	lwarpmk	99, 173
pdfseparate (program) [requirement]	42, 46	Mac OS	68, 109
pdfsync (package)	483	Madcap	40
pdfcairo (program) [requirement]	42, 46	MathJax	84, 85, 320
pdftotext (program) [requirement]	42, 46	MS-Windows	68, 109
Perl	47	OpenOffice	39
perl (program) [requirement]	47	Pandoc	39
\phantomsection	7388	Plastex	38
picture (environment)	361, 6925	\TeX2page	38
placeins (package)	484	\TeX4ht	38
Plastex (program)	38	TtH	38
\popclose	3163	Unix	68, 109
Poppler	42, 46	Windows	68, 109
		Word	39

xindy	72
project.css (file)	72
project.lwarpmkconf (file)	134
project_html.tex (file)	133
published	247
\pushclose	3136
Q	
\quadquad	378, 7301
\quad	378, 7296
quote (environment)	4576
R	
ragged2e (package)	484
\raggedbottom	3309
\raggedleft	236
\raggedright	236
\raisebox	7111
realscripts (package)	485
\ref	5523
refcount (package)	122
\reflectbox	6789
relsize (package)	486
[requirement]:	
LuaTeX (program)	42
MathJax (program)	42
pdfcrop (program)	42
pdfLaTeX (program)	42
pdfseparate (program)	42, 46
pdftocairo (program)	42, 46
pdftotext (program)	42, 46
perl (program)	47
XeLaTeX (program)	42
\RequirePackage	388
\resizebox	6794
\restylefloat	17
\ResumeTabular	5279
\rmfamily	7223
romanbar (package)	487
romanbarpagenumber (package)	488
\rotatebox	6759
rotating (package)	488
rotfloat (package)	488
\rule	7385
S	
sample_project.css (file)	72, 170
savetrees (package)	489
\sb	7231
\scalebox	6776
\scaleffnt (package)	489
\scshape	7228
section	
depths	104
missing	58
\section	4180
\SetHTMLFileNumber	3219
setspace (package)	490
settings	
css project-specific	72
CSS selection	72
HTML conversion	57
lwarp package options	67
selecting output	69
title page	70
\sffamily	7224
\sfrac	543
shadow (package)	491
showidx (package)	491
showkeys (package)	491
sidecap (package)	492
sidenotes (package)	493
SideTOCDepth (counter)	58, 313
\sidetocname	5778
siunitx (package)	85, 344, 494
\sloppy	3311
\soul (package)	495
\sp	7230
space	
horizontal	73
between minipages	378
split	
miss-numbered	83, 85, 319, 472
stack depths	104
\StartDefiningTabulars	4787
subequations	
MathJax	85, 320
subfig (package)	496
\subparagraph	4213
\subsection	4188
\subsubsection	4196
subtitle	247
supertabular (package)	502
SVG	
images showing as HTML	54, 102
math summary	84, 319
mathsvg option	111
syntonly (package)	503

T	U		
\tableofcontents	60	tikz (package)	512
tabular		tikzpicture (environment)	8
baseline	271	\title	70
HTML columnn conversion	272	titlepage	
macros inside	89, 259	subtitle and published	247
misplaced \noalign	89, 260	titlepage (environment)	70, 4392
misplaced alignment tab character &		titleps (package)	513
.	88, 258	titleref (package)	515
multicolumn with multirow	468	titlesec (package)	515
row corruption	89, 259	titletoc (package)	516
\TabularMacro	5277	titling (package)	517
tabularx (package)	504	titlingpage (environment)	14, 70
tabulary (package)	504	[tocbibind]:	
tagged equations		numindex (option)	83
Mathjax	85, 320	tocbibind (package)	83, 521
\TeX	7393	tocdepth (counter)	57
\TeX2page (program)	38	tocloft (package)	522
\TeX4ht (program)	38	\tracingl warp	103, 495
textarea (package)	505	transparent (package)	527
\textbf	7152	trivfloat	
\textcolor	33	with newfloat, algorithmic	528
\textcomp (package)	67, 505	trivfloat (package)	527
\textgreater	3209	troubleshooting	101
\textit	7167	HTML debug comments	128
\textless	3209	\ttfamily	7225
\textmd	7149	TtH (program)	38
\textnormal	7176	tutorial.tex (file)	48
\textpos (package)	507	typearea (package)	528
\textrm	7155		
\textsc	7170	ulem (package)	529
\textsf	7158	underscore	
\textsl	7173	filename	57, 74
\textsubscript	7233	Unicode	
\textsuperscript	7232	enhanced coverage	66
\texttt	7161	file & section names	222
\textup	7164	input characters	107
\tfl@chapter@fix	528	selection	66
\thanks	71	units (package)	86
\thanksmarkseries	84	Unix (program)	68, 109
\theauthor	240	\up	7234
\thedate	240	upref (package)	530
theindex (environment)	5856	\upshape	7226
theorem (package)	508	\url	5575
\thetitle	240	usingOSWindows (boolean)	109
\thispagestyle	3306	UTF-8	
threeparttable (package)	511	enhanced coverage	66
tikz		file & section names	222
catcodes	512	index	67
dollar redefined	512	selection	66

utility		Word (program)	39	
programs	41		
		word processor		
		HTML conversion settings	96, 97, 129	
		wrapfig (package)	533	
			X	
V				
Verbatim (environment)	13		
verbatim (environment)	4633		
\VerbatimHTMLWidth (length)	251	xcolor (package)	534
\verbatiminput	4625	xdyFilename (option)	73, 111
verse (environment)	2	XeLaTeX	
verse (package)	531, 532	detection	118
viewport			file & section names	222
HTML meta tag		233	\XeLaTeX	7406
vleftmargini (length)	92, 250, 531	XeLaTeX (program) [requirement]	42
vleftskip (length)	92, 250, 531, 532	\XeTeX	7406
\vspace	7334	xfrac (package)	123, 542
			\xfracHTMLfontsize	3
			xifthen (package)	121
			xindy	
			customizing	72
W			xindy (program)	72
wallpaper (package)	532	xltextra (package)	545
warning icon	106	xmpincl (package)	545
warpall (environment)	69, 100	xparse (package)	120
warpHTML (environment)	50, 61, 69, 101	xstring (package)	121
warpHTML (option)	67, 110	xtab (package)	546
\warpHTMLonly	61, 69, 90		
warpingHTML (boolean)	110	Z	
warpingprint (boolean)	110		
warpprint (environment)	60, 69, 101	zref (package)	122
warpprint (option)	67, 110	zwpagelayout (package)	547
\warpprintonly	61, 69, 89		
Windows (program)	68, 109		