



The **lwarp** package

ℒ_ṼṼ_Ṽ to HTML

v0.41 — 2017/10/07

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

The **lwarp** package allows ℒ_ṼṼ_Ṽ 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 pdfℒ_ṼṼ_Ṽ, Luaℒ_ṼṼ_Ṽ, or X_Ṽℒ_ṼṼ_Ṽ. A `texlua` script removes the need for system utilities such as `make` and `gawk`, and also supports `xindy` and `latexmk`. Configuration is automatic at the first manual compile.

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

A modular package-loading system uses the **lwarp** version of a package for HTML when available. More than a hundred ℒ_ṼṼ_Ṽ 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 **lwarp, and some things may change in response to user feedback and further project development.**

License:

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in <http://www.latex-project.org/lppl.txt> and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

Contents

1	lwarp.sty	29
1	Updates	29
2	Introduction	36
2.1	Supported packages and features	37
3	Alternatives	41
3.1	Internet class	41
3.2	TeX4ht	41
3.3	Translators	41
3.4	AsciiDoc	42
3.5	Pandoc	42
3.6	Word processors	42
3.7	Commercial systems	43
3.8	Comparisons	43
4	Installation	44
4.1	Installing the lwarp package	44
4.2	Installing the lwarpmk utility	47
4.2.1	Using a local copy of lwarpmk	48
4.3	Installing additional utilities	49
5	Tutorial	51
5.1	Starting a new project	51
5.2	Compiling the print version with lwarpmk	55
5.3	Compiling the HTML version with lwarpmk	56
5.4	Generating the SVG images	57
5.5	Using MathJax for math	58
5.6	Changing the CSS style	59
5.7	Customizing the HTML output	60
5.8	Using latexmk	65
5.9	Using XeLaTeX or LuaLaTeX	66
5.10	Using a glossary	67
5.11	Cleaning auxiliary files	68
5.12	Cleaning auxiliary and output files	68
5.13	Processing multiple projects in the same directory	68
5.14	Using the make utility	68
6	Additional details	69
6.1	Font and UTF-8 support	69
6.1.1	Indexes and UTF-8	70
6.2	lwarp package loading and options	70
6.3	Selecting the operating system	71

6.4	Selecting actions for print or HTML output	72
6.5	Commands to be placed into the <code>warpprint</code> environment	73
6.6	Title page	73
6.7	HTML page meta descriptions	74
6.8	HTML page meta author	74
6.9	CSS	75
6.10	Modifying <code>xindy</code> index processing	75
7	Special cases and limitations	76
7.1	Formatting	76
7.1.1	Text formatting	76
7.1.2	Horizontal space	76
7.1.3	Text alignment	76
7.1.4	Accents	77
7.1.5	Textcomp	77
7.1.6	Superscripts and other non-math uses of math mode	77
7.1.7	Empty <code>\item</code> followed by a new line of text or a nested list:	77
7.1.8	Filenames in lists	77
7.1.9	<code>relsize</code> package	77
7.2	Boxes and minipages	78
7.2.1	Save Boxes	78
7.2.2	Minipages	78
7.2.3	Side-by-side minipages	79
7.2.4	Framed minipages and other environments	79
7.2.5	<code>fancybox</code> package	80
7.2.6	<code>mdframed</code> package	81
7.3	Cross-references	82
7.3.1	Page references	82
7.3.2	<code>cleveref</code> and <code>varioref</code> packages	82
7.3.3	Hyperlinks, <code>hyperref</code> , and <code>url</code>	83
7.3.4	Footnotes and page notes	83
7.4	Front and back matter	83
7.4.1	Starred chapters and sections	83
7.4.2	<code>abstract</code> package	84
7.4.3	<code>titling</code> and <code>authblk</code>	84
7.4.4	<code>tocloft</code> package	84
7.4.5	<code>appendix</code> package	85
7.4.6	<code>pagenote</code> package	85
7.4.7	<code>endnotes</code> package	85
7.4.8	<code>glossaries</code> package	85
7.4.9	Index and the <code>tocbibind</code> package	86
7.5	Math	87
7.5.1	<code>ntheorem</code> package	89
7.5.2	<code>siunitx</code> package	89
7.5.3	<code>units</code> and <code>nicefrac</code> packages	89
7.5.4	<code>newtxmath</code> package	89

7.6	Graphics	90
7.6.1	grffile package	91
7.6.2	color package	91
7.6.3	xcolor package	91
7.7	Tabular	92
7.7.1	longtable package	94
7.7.2	supertabular and xtab packages	95
7.7.3	bigdelim package	95
7.8	Floats	96
7.8.1	float, trivfloat, and/or algorithmicx together	96
7.8.2	caption and subcaption packages	96
7.8.3	subfig package	96
7.8.4	floatrow package	96
7.9	Miscellaneous	97
7.9.1	verse and memoir	97
7.9.2	newclude package	97
7.9.3	babel package	98
8	EPUB conversion	99
9	Word-processor conversion	101
10	Modifying lwarp	103
10.1	Creating an lwarp version of a package	103
10.2	Adding a package to the lwarp.dtx file	104
10.3	Testing lwarp	104
10.4	Modifying lwarpmk	104
11	Troubleshooting	106
11.1	Using the lwarp.sty package	106
11.1.1	Debug tracing output	108
11.2	Compiling the lwarp.dtx file	108
12	Implementation	109
13	Stack depths	109
14	Source Code	111
15	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	112
16	Unicode Input Characters	112
17	Early package requirements	113
18	Operating-System portability	114
18.1	Common portability code	114

18.2	Unix, Linux, and Mac OS	114
18.3	MS-Windows	114
19	Package options	115
19.1	Conditional compilation	118
20	Misplaced packages	119
21	Required packages	123
22	Loading packages	128
23	File handles	131
24	Include a file	131
25	Copying a file	132
26	Debugging messages	133
27	HTML-conversion output modifications	134
28	Remembering original formatting macros	135
29	Accents	136
30	Configuration Files	138
30.1	project_html.tex	138
30.2	lwarpmk.conf	138
30.3	project.lwarpmkconf	139
30.4	lwarp.css	140
30.5	lwarp_sagebrush.css	166
30.6	lwarp_formal.css	171
30.7	sample_project.css	176
30.8	lwarp.xdy	176
30.9	lwarp_mathjax.txt	177
30.10	lwarpmk option	179
31	Stacks	190
31.1	Assigning depths	191
31.2	Closing actions	191
31.3	Closing depths	192
31.4	Pushing and popping the stack	192
32	Data arrays	194
33	HTML entities	194

34	HTML filename generation	195
35	Homepage link	198
36	\PrintStack diagnostic tool	198
37	Closing stack levels	199
38	PDF pages and styles	199
39	HTML tags, spans, divs, elements	201
39.1	Mapping \LaTeX Sections to HTML Sections	201
39.2	Babel-French	201
39.3	HTML tags	202
39.4	Block tags and comments	205
39.5	Div class and element class	206
39.6	Single-line elements	207
39.7	HTML5 semantic elements	207
39.8	High-level block and inline classes	208
39.9	Closing HTML tags	209
40	Paragraph handling	210
41	Paragraph start/stop handling	213
42	Page headers and footers	216
43	CSS	217
44	HTML meta description and author	218
45	Footnotes	218
45.1	Regular page footnotes	219
45.2	Minipage footnotes	219
45.3	Titlepage thanks	219
45.4	Regular page footnote implementation	219
45.5	Minipage footnote implementation	221
45.6	Printing pending footnotes	222
46	Marginpars	222
47	Splitting HTML files	223
48	Sectioning	228
48.1	User-level starred section commands	228
48.2	Book class commands	229
48.3	Sectioning support macros	230
48.4	\section and friends	236

49	Starting a new file	238
50	Starting HTML output	241
51	Ending HTML output	244
52	Title page	245
52.1	Setting the title, etc.	246
52.2	Changes for <code>\affiliation</code>	247
52.3	Printing the thanks	248
52.4	Printing the title, etc. in HTML	248
52.5	Printing the title, etc. in print form	250
52.6	<code>\maketitle</code> for HTML output	250
52.7	<code>\published</code> and <code>\subtitle</code>	253
53	Abstract	254
54	Quote and verse	255
54.1	Citations and attributions	255
54.2	Quotes, quotations	256
54.3	Verse	256
55	Verbatim	257
56	Theorems	259
57	Lists	260
57.1	Itemize	261
57.2	Enumerate	262
57.3	Description	262
58	Tabular	264
58.1	Limitations	264
58.2	Token lookahead	267
58.3	Booleans	268
58.4	Handling <code>&</code> , <code>@</code> , <code>!</code> , and <code>bar</code>	269
58.4.1	Localizing <code>&</code> catcodes	271
58.4.2	Handling <code>&</code>	271
58.4.3	Filling an unfinished row	272
58.5	Handling <code>\\</code>	273
58.6	Variables	274
58.7	Parsing <code>@</code> , <code>></code> , <code><</code> , <code>!</code> , <code>bar</code> columns	276
58.8	Parsing <code>'l'</code> , <code>'c'</code> , or <code>'r'</code> columns	278
58.9	Parsing <code>'p'</code> , <code>'m'</code> , or <code>'b'</code> columns	279
58.10	Parsing <code>'D'</code> columns	279
58.11	Parsing the column specifications	280
58.12	Starting a new row	284

58.13	Printing vertical bar tags	285
58.14	Printing at or bang tags	285
58.15	Data opening tag	286
58.16	Midrules	288
58.17	Multicolumns	291
58.17.1	Parsing multicolumns	291
58.17.2	Multicolumn factored code	294
58.17.3	Multicolumn	295
58.17.4	Longtable captions	296
58.17.5	Counting HTML tabular columns	298
58.18	Multicolumnrow	300
58.19	Utility macros inside a table	301
58.20	Special-case tabular markers	301
58.21	Checking for a new table cell	303
58.22	\mrowcell	305
58.23	\mcolrowcell	305
58.24	New \tabular definition	305
58.25	Array	310
59	Cross-references	310
59.1	Setup	310
59.2	Zref setup	312
59.3	Labels	314
59.4	References	315
59.5	Hyper-references	317
60	Floats	320
60.1	Float captions	321
60.1.1	Caption inside a float environment	322
60.1.2	Caption and LOF linking and tracking	323
61	Table of Contents, LOF, LOT	326
61.1	Reading and printing the TOC	326
61.2	High-level TOC commands	328
61.3	Side TOC	329
61.4	Low-level TOC line formatting	330
62	Index and glossary	333
63	Math	335
63.1	Inline and display math	337
63.2	MathJax support	339
63.3	Equation environment	343
63.4	AMS Math environments	345
63.4.1	Support macros	345
63.4.2	Environment patches	346

64	Lateximages	353
65	center, flushleft, flushright	359
66	Pre-loaded packages	360
67	Siunitx	361
68	Graphics	361
68.1	xcolor print-mode additions	362
69	Cleveref	366
70	Picture	368
71	Boxes and Minipages	368
71.1	Counters and lengths	369
71.2	Footnote handling	369
71.3	Minipage handling	370
71.4	Parbox, makebox, framebox, fbox, raisebox	373
72	Direct formatting	378
73	Skips, spaces, font sizes	383
74	\phantomsection	390
75	\LaTeX and other logos	390
75.1	HTML logos	391
75.2	Print logos	392
76	\AtBeginDocument, \AtEndDocument	393
77	Trademarks	394
2	lwarp-a4.sty	395
78	a4	395
3	lwarp-a4wide.sty	395
79	a4wide	395
4	lwarp-a5comb.sty	395
80	a5comb	395

5	lwarp-abstract.sty	395
81	abstract	395
6	lwarp-adjmulticol.sty	397
82	adjmulticol	397
7	lwarp-addlines.sty	398
83	addlines	398
8	lwarp-afterpage.sty	398
84	afterpage	398
9	lwarp-algorithmicx.sty	399
85	algorithmicx	399
10	lwarp-alltt.sty	399
86	alltt	399
11	lwarp-amsthm.sty	400
87	amsthm	400
12	lwarp-anonchap.sty	402
88	anonchap	402
13	lwarp-anysize.sty	403
89	anysize	403
14	lwarp-appendix.sty	403
90	appendix	403

15	lwarp-arabicfront.sty	404
91	arabicfront	404
16	lwarp-authblk.sty	404
92	authblk	404
17	lwarp-balance.sty	405
93	balance	405
18	lwarp-bigdelim.sty	405
94	bigdelim	405
19	lwarp-bigstrut.sty	407
95	bigstrut	407
20	lwarp-bookmark.sty	407
96	bookmark	407
21	lwarp-booktabs.sty	407
97	booktabs	407
22	lwarp-boxedminipage.sty	408
98	boxedminipage	408
23	lwarp-boxedminipage2e.sty	408
99	boxedminipage2e	408
24	lwarp-caption2.sty	409
100	caption2	409

25	lwarp-ccaption.sty	409
101	ccaption	409
26	lwarp-changepage.sty	409
102	changepage	409
27	lwarp-chngpage.sty	409
103	chngpage	409
28	lwarp-chappg.sty	410
104	chappg	410
29	lwarp-color.sty	410
105	color	410
30	lwarp-crop.sty	410
106	crop	410
31	lwarp-cuted.sty	411
107	cuted	411
32	lwarp-cutwin.sty	411
108	cutwin	411
33	lwarp-dblfnote.sty	412
109	dblfnote	412
34	lwarp-dcolumn.sty	412
110	dcolumn	412

35	lwarp-draftwatermark.sty	412
111	draftwatermark	412
36	lwarp-ebook.sty	413
112	ebook	413
37	lwarp-ellipsis.sty	413
113	ellipsis	413
38	lwarp-emptypage.sty	414
114	emptypage	414
39	lwarp-endnotes.sty	414
115	endnotes	414
40	lwarp-enumerate.sty	415
116	enumerate	415
41	lwarp-enumitem.sty	415
117	enumitem	415
42	lwarp-epigraph.sty	416
118	epigraph	416
43	lwarp-eso-pic.sty	417
119	eso-pic	417
44	lwarp-everypage.sty	417
120	everypage	417

45	lwarp-everyshi.sty	418
121	everyshi	418
46	lwarp-extramarks.sty	418
122	extramarks	418
47	lwarp-fancybox.sty	418
123	fancybox	418
48	lwarp-fancyhdr.sty	422
124	fancyhdr	422
49	lwarp-fancyvrb.sty	423
125	fancyvrb	423
50	lwarp-figcaps.sty	428
126	figcaps	428
51	lwarp-fix2col.sty	429
127	fix2col	429
52	lwarp-float.sty	429
128	float and \newfloat	429
53	lwarp-floatflt.sty	430
129	floatflt	430
54	lwarp-floatpag.sty	431
130	floatpag	431

55	lwarp-floatrow.sty	431
131	floatrow	431
56	lwarp-flushend.sty	437
132	flushend	437
57	lwarp-fncychap.sty	437
133	fncychap	437
58	lwarp-fnpos.sty	438
134	fnpos	438
59	lwarp-fontenc.sty	438
135	fontenc	438
60	lwarp-fontspec.sty	439
136	fontspec	439
61	lwarp-footmisc.sty	439
137	footmisc	439
62	lwarp-footnote.sty	440
138	footnote	440
63	lwarp-footnotehyper.sty	441
139	footnotehyper	441
64	lwarp-framed.sty	441
140	framed	441

65	lwarp-ftnright.sty	444
141	ftnright	444
66	lwarp-fullpage.sty	444
142	fullpage	444
67	lwarp-fullwidth.sty	444
143	fullwidth	444
68	lwarp-geometry.sty	445
144	geometry	445
69	lwarp-glossaries.sty	445
145	glossaries	445
70	lwarp-graphics.sty	447
146	graphics	447
146.1	Graphics extensions	447
146.2	Length conversions and graphics options	448
146.3	\includegraphics	451
71	lwarp-graphicx.sty	457
147	graphicx	457
72	lwarp-grffile.sty	457
148	grffile	457
73	lwarp-hyperref.sty	457
149	hyperref	457

74	lwarp-hyperxmp.sty	462
150	hyperxmp	462
75	lwarp-idxlayout.sty	462
151	idxlayout	462
76	lwarp-indentfirst.sty	463
152	indentfirst	463
77	lwarp-inputenc.sty	463
153	inputenc	463
78	lwarp-keyfloat.sty	463
154	keyfloat	463
79	lwarp-layout.sty	464
155	layout	464
80	lwarp-letterspace.sty	465
156	letterspace	465
81	lwarp-lettrine.sty	465
157	lettrine	465
82	lwarp-lips.sty	466
158	lips	466

83	lwarp-listings.sty	467
159	listings	467
84	lwarp-longtable.sty	470
160	longtable	470
85	lwarp-lscape.sty	472
161	lscape	472
86	lwarp-ltcaption.sty	472
162	ltcaption	472
87	lwarp-marginfit.sty	473
163	marginfit	473
88	lwarp-marginfix.sty	473
164	marginfix	473
89	lwarp-marginnote.sty	473
165	marginnote	473
90	lwarp-mcaption.sty	474
166	mcaption	474
91	lwarp-mdframed.sty	474
167	mdframed	474

92	lwarp-metalogo.sty	483
168	metalogo	483
93	lwarp-microtype.sty	483
169	microtype	483
94	lwarp-midfloat.sty	484
170	midfloat	484
95	lwarp-moreverb.sty	484
171	moreverb	484
96	lwarp-mparhack.sty	486
172	mparhack	486
97	lwarp-multicol.sty	486
173	multicol	486
98	lwarp-multirow.sty	487
174	multirow	487
174.1	Multirow	488
174.2	Combined multicolumn and multirow	490
99	lwarp-nameref.sty	491
175	nameref	491
100	lwarp-needspace.sty	491
176	needspace	491

101	lwarp-newclude.sty	491
177	newclude	491
102	lwarp-newunicodechar.sty	492
178	newunicodechar	492
103	lwarp-nextpage.sty	492
179	nextpage	492
104	lwarp-nonumonpart.sty	492
180	nonumonpart	492
105	lwarp-nopageno.sty	493
181	nopageno	493
106	lwarp-nowidow.sty	493
182	nowidow	493
107	lwarp-ntheorem.sty	493
183	ntheorem	493
108	lwarp-pagenote.sty	504
184	pagenote	504
109	lwarp-paralist.sty	504
185	paralist	504
110	lwarp-parskip.sty	505
186	parskip	505

111	lwarp-pdfscape.sty	505
187	pdfscape	505
112	lwarp-pdfsync.sty	505
188	pdfsync	505
113	lwarp-pfnote.sty	506
189	pfnote	506
114	lwarp-placeins.sty	506
190	placeins	506
115	lwarp-prelim2e.sty	506
191	prelim2e	506
116	lwarp-quotchap.sty	507
192	quotchap	507
117	lwarp-ragged2e.sty	508
193	ragged2e	508
118	lwarp-realscripts.sty	509
194	realscripts	509
119	lwarp-relsize.sty	510
195	relsize	510

120	lwarp-romanbar.sty	511
196	romanbar	511
121	lwarp-romanbarpagenumber.sty	511
197	romanbarpagenumber	511
122	lwarp-rotating.sty	512
198	rotating	512
123	lwarp-rotfloat.sty	512
199	rotfloat	512
124	lwarp-savetrees.sty	513
200	savetrees	513
125	lwarp-scalefnt.sty	513
201	scalefnt	513
126	lwarp-sectsty.sty	513
202	sectsty	513
127	lwarp-setspace.sty	514
203	setspace	514
128	lwarp-shadow.sty	515
204	shadow	515
129	lwarp-showidx.sty	516
205	showidx	516

130	lwarp-showkeys.sty	516
206	showkeys	516
131	lwarp-sidecap.sty	516
207	sidecap	516
132	lwarp-sidenotes.sty	517
208	sidenotes	517
133	lwarp-siunitx.sty	519
209	siunitx	519
134	lwarp-soul.sty	519
210	soul	519
135	lwarp-stabular.sty	521
211	stabular	521
136	lwarp-subfig.sty	521
212	subfig	521
137	lwarp-supertabular.sty	527
213	supertabular	527
138	lwarp-syntonly.sty	528
214	syntonly	528
139	lwarp-tabls.sty	529
215	tabls	529

140	lwarp-tabularx.sty	529
216	tabularx	529
141	lwarp-tabulary.sty	530
217	tabulary	530
142	lwarp-textarea.sty	530
218	textarea	530
143	lwarp-textcomp.sty	531
219	textcomp	531
144	lwarp-textpos.sty	534
220	textpos	534
145	lwarp-theorem.sty	534
221	theorem	534
146	lwarp-threeparttable.sty	538
222	threeparttable	538
147	lwarp-tikz.sty	539
223	tikz	539
148	lwarp-titleps.sty	540
224	titleps	540
149	lwarp-titleref.sty	543
225	titleref	543

150	lwarp-titlesec.sty	543
226	titlesec	543
151	lwarp-titletoc.sty	545
227	titletoc	545
152	lwarp-titling.sty	546
228	titling	546
153	lwarp-tocbibind.sty	551
229	tocbibind	551
154	lwarp-tocloft.sty	553
230	tocloft	553
155	lwarp-transparent.sty	558
231	transparent	558
156	lwarp-trivfloat.sty	559
232	trivfloat	559
232.1	Combining \newfloat, \trivfloat, and algorithmicx	560
157	lwarp-typearea.sty	560
233	typearea	560
158	lwarp-ulem.sty	560
234	ulem	560

159	lwarp-upref.sty	562
235	upref	562
160	lwarp-verse.sty	562
236	verse	562
161	lwarp-wallpaper.sty	564
237	wallpaper	564
162	lwarp-wrapfig.sty	565
238	wrapfig	565
163	lwarp-xcolor.sty	566
239	xcolor	566
239.1	Xcolor definitions: location and timing	567
239.2	Code	569
164	lwarp-xfrac.sty	574
240	xfrac	574
165	lwarp-xltxtra.sty	576
241	xltxtra	576
166	lwarp-xmpinl.sty	577
242	xmpinl	577
167	lwarp-xtab.sty	577
243	xtab	577

lwarp	27
-------	----

168	lwarp-zwpagelayout.sty	579
------------	-------------------------------	------------

244	zwpagelayout	579
------------	---------------------	------------

	Change History and Index	581
--	---------------------------------	------------

List of Figures

1	tutorial.tex listing	52
---	--------------------------------	----

List of Tables

1	TeX-HTML generation — lwarp package — Supported functions . .	37
2	Required software programs	45
3	Files created along with the print version	54
4	Package options	71
5	Section depths and HTML headings	110
6	Tabular baseline	280
7	Tabular HTML column conversions	281
8	Cross-referencing data structures	311
9	Float data structures	320

File 1 **lwarp.sty****1 Updates**

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

For a detailed list of changes, see the Change History on page 581.

v0.41:

tabular

 new syntax

- Added tabular vertical rules, subject to some limitations. See the rules section of section 7.7.
- Improved booktabs: Width and trim are honored.
- Added `\mcolrowcell` for empty cells inside a `\multicolumnrow`. Use `\mcolrowcell` instead of `\mrowcell` for two-dimensional cells created by `\multicolumnrow`. Continue to use `\mrowcell` for empty cells in a `\multirow`. See section 174.2 on page 490.
- Fix: Unfinished tabular rows are automatically filled.
- Fix for tabular column specifiers while using babel-french. (`\NoAutoSpacing` is activated then nullified inside the tabular, due to a conflict with the tabular column parsing code.)

v0.40:

graphics, graphicx

`\includegraphics path` image file extensions

bigdelim

symbols

fixes

- graphics and graphicx have been moved from the lwarp core, and are only loaded if requested with `\usepackage`.
- Improved graphics `\graphicspath` support. Multiple image directories may now be used. Refer to `.pdf files without a file extension` to allow the HTML version to use a `.svg`, `.png`, `.jpg`, or `.gif` version instead. See section 7.6.
- grffile is now directly supported instead of emulated.
- Fix for bigdelim, and improved documentation. See section 94.
- Improved \TeX and textcomp symbols.
- Fix for \TeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for xltextra with \TeX .
- Fixes for tocbibind with `\simplechapter`, etc.

- Fixes for `\multicolumnrow` and `\nullfonts` with older versions of multirow and xparse.
- Added `\underline`.
- Added `adjmulticol`.
- Added `cuted`, `midfloat`.
- Added `pfnote`, `fnpos`, `dblfnote`.
- Added `stabular`, `tabls`.
- Added `sectsty`, `anonchap`, `quotchap`.

v0.39:

- 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 174.2.
- 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:

- 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.
- Modified the tutorial to use the new `\ForceHTMLPage` and `\ForceHTMLTOC` macros.
- Added `appendix`, `tocbibind`, `fncychap`, `fix2col`.
- Added `resize`, `scalefnt`.
- Added `realscripts`, `metalogo`, `xltxtra`.

page numbering

front & back matter

- Added grffile, romanbar.
- Added arabicfront, chappg, nonumonpart, nopageno, romanbarpagenumber.
- Docs: Improved description of the use of front/back matter. See section 7.4.
- Fix: color requests xcolor.
- Fix: \part for article class.

v0.37:

\include for HTML
latexmk

accents and symbols
babel-french

- \include now maintains independent .aux files for HTML versions.
- comment, used by lwarp, now maintains independent cut files for print and HTML versions, helping latexmk to better know whether to recompile.
- Improved support for \LaTeX accents, textcomp, siunitx symbols.
- Improved babel-french handling for load order and ~ tilde.

v0.36:

boxes and frames

babel-french

footnotes

siunitx

- Recorganized the documentation section regarding special cases and limitations. (Section 7)
- Improved source formatting.
- \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: Improvements for French variants, load order, footnotes, ellipses.
- Improved footnote numbering. lateximage footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with MathJax.
- Improved siunitx units.
- Fix for filenames while using MathJax.
- Fix for \rule when xcolor is not loaded.
- Added transparent, upref.

v0.35: Fix: \textbf and related.

v0.34:

 Optional arguments

- BlockClass's optional argument has been moved in front of the mandatory argument:

BlockClass[style]{class} (NEW)

instead of:

BlockClass{class}[style] (OLD)

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

⚠ Optional arguments

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

```
\InlineClass[style]{class}{text}
```

spans with minipages

framing minipages

- Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML formatting. 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`.

eqnarray

- Improved font control for `lateximages` and SVG math.

verbatim packages

- Added the `eqnarray` environments.
- `fancyvrb` is no longer required (preloaded), but is still supported.
- Added `verbatim` and `moreverb`.

framing packages

list packages

- 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`.

babel-french

- Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
- Adds fixed-width HTML spaces around punctuation when using `babel-french`. Lua_T_EX does not yet use the extra punctuation spacing.

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
<code>\HomeHTMLFileName</code>	<code>HomeHTMLFilename</code>
<code>\HTMLFileName</code>	<code>HTMLFilename</code>
<code>\useLatexmk</code>	<code>latexmk</code>
<code>\warpOSWindows</code>	<code>OSWindows</code>

Old Package Option	New Package Option
<code>lwarpmklang</code> (new)	<code>IndexLanguage</code> <code>xdyFilename</code>

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

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


```
\usepackage[
  HomeHTMLFilename=index,
  HTMLFilename={},
  IndexLanguage=english
]{lwarp}
```
- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
 1. Make the changes shown above.
 2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
 3. The old file `lwarp_html.xdy` may be deleted.
- The new lwarp package option `xdyFilename` may be used to tell `lwarpmk` to use a custom `.xdy` file instead of `lwarp.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 `lwarp-newproject` and `lwarp`. Sets the language to use while processing the glossary. (As of v0.30, this has been changed to the `IndexLanguage` option.)
- 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 placed 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 regareding 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 \LaTeX version. These `lwarp-packagename.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 37.)

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 \LaTeX 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 \LaTeX code.

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

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

lwarp 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> , <code>sectsty</code> , <code>anonchp</code> , <code>quotchap</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 HTML ALT tags containing the \LaTeX source for the math expression. MathJax supported as an alternative. \mathcal{AMS} environments are supported. User-defined macros are available during conversion, due to native \LaTeX processing.
Theorems:	Support for native \LaTeX 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>

lwarp Supported Functions — continued

Category	Status
Floats:	Appear where declared. float, rotfloat, newfloat, caption and subcaption, subfig, capt-of, placeins, trivfloat, floatrow, subfloat, keyfloat, wrapfig, cutwin, floatflt.
Tabular	tabular environment, array, tabularx, tabulary, threeparttable, multirow, longtable, supertabular, xtab, booktabs.
Graphics	graphics and graphicx are emulated. \includegraphics supports width, height, origin, angle, and scale tags, and adds class. References to PDF files are changed to SVG, other image types are accepted as well. \rotatebox and \scalebox are supported as well as HTML can handle. grffile is supported.
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 mixing . <code>\textcolor</code> , <code>\colorbox</code> , <code>\fcolorbox</code> . Enhanced for HTML compatibility.
Lists:	Standard \TeX environments are supported, along with enumitem, enumerate, paralist. Spacing is still controlled by CSS.
Environments:	Standard \TeX environments are supported.
minipage:	Supported with some HTML-imposed limitations. Nested minipages are supported. Footnotes appear at the bottom of the HTML page.
Quotations:	verse, csquotes, epigraph
Verbatim:	verbatim, moreverb, fancyvrb (except for verbatim footnotes).
Frames:	framed, fancybox, mdframed, boxedminipage2e, shadow.

lwarp Supported Functions — continued

Category	Status
multicol:	Emulated, with css. Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
fullwidth:	Emulated.
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are not yet supported. <code>lettrine</code> , <code>ulem</code> , <code>soul</code> , <code>relsize</code> , <code>scaletnt</code> , and <code>realscripts</code> are supported.
Ordinals:	<code>nth</code> , <code>fmtcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for <code>f</code> , <code>q</code> , <code>t</code> 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 <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
Fonts:	Used as-is. Appear in math expressions or embedded image environments.
Symbols:	Native \TeX diacriticals, <code>textcomp</code> .
Working as-is:	Various utility, calculation, file, and text-only packages, such as <code>calc</code> , <code>fileerr</code> , <code>somedefs</code> , <code>trace</code> , <code>xspace</code> .
Where:	
Supported:	The existing \TeX package is used.
Emulated:	The \TeX 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, lwarp may be a good choice. For new projects with a large number of documents, it may be worth investigating the alternatives before decided which path to take.

3.1 Internet class

Cls internet The closest to lwarp 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, lwarp 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 **L^AT_EXML**: <http://dlmf.nist.gov/LaTeXML/>

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

Prog LaTeX2HTML **L^AT_EX2HTML**: <http://www.latex2html.org/>
and <http://ctan.org/pkg/latex2html>.

Prog TeX2page **T_EX2page**: <http://ds26gte.github.io/tex2page/index.html>

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

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

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

Asciidoctor-LaTeX:

Prog Asciidoctor-LaTeX <http://www.noteshare.io/book/asciidoctor-latex-manual>
<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 years in their abilities to represent math with a \TeX -ish input syntax, unicode math fonts, and high-quality output, and also generate HTML with varying success. See recent developments in Microsoft® Word® and LibreOffice™ Writer.

Prog LibreOffice

Prog OpenOffice

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	FrameMaker	
Prog	InDesign	
	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, \LaTeX spells things out in readable words but takes longer to type, although integrated editors exist which can provide faster entry and a graphic user interface. For those functions which are covered by the typical markup language it is arguable that \LaTeX is comparably easy to learn, while \LaTeX provides many more advanced features where needed, along with a large number of pre-existing packages which provide solutions to numerous common tasks.

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

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

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

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 lwarp package

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

Pre-installed: Try entering into a command line:

```
Enter ⇒ kpsewhich lwarp.sty
```

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

\TeX Live: If using a \TeX Live distribution, try installing via `tlmgr`:

```
Enter ⇒ tlmgr install lwarp
```

MiK \TeX : If using MiK \TeX , try using the package installer to install the package `lwarp`. Also update the package `miktex-misc`, which will install the `lwarpmk` executable.

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

CTAN TDS archive: lwarp may be downloaded from the Comprehensive \TeX Archive:

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

\TeX Live:

```
Enter ⇒ kpsewhich -var-value TEXMFLOCAL
```

MiK \TeX :

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 \LaTeX distribution:

From \TeX Live: <http://tug.org/texlive/>.

\LaTeX : pdf \LaTeX , x \LaTeX , or lua \LaTeX .

The lwarp package: This package.

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

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

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

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

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

Poppler PDF utilities:

pdftotext: Used to convert PDF to text.

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

pdftocairo: Used to convert images to SVG.

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

From Poppler: poppler.freedesktop.org.

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

```
Enter ⇒ brew install poppler
```

For Windows, see:

<https://sourceforge.net/projects/poppler-win32/> and:
<http://blog.alivate.com.au/poppler-windows/>

Perl:

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

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 MiKTeX, start the program called MiKTeX 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 lwarp package.
2. Download the zip archive lwarp.zip into your own lwarp directory.
3. Unpack lwarp.zip.
4. Locate the contents lwarp.dtx and lwarp.ins
5. Create the documentation:
 Enter ⇒ `pdflatex lwarp.dtx`
 (several times)
6. Create the .sty files:
 Enter ⇒ `pdflatex lwarp.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/lwarp
8. Copy the documentation lwarp.pdf to a source directory in the local tree, such as:
 <texlocal>/doc/local/lwarp
9. Renew the cache:
 Enter ⇒ `mktxlsr`
 — or —
 Enter ⇒ `texhash`
 Or, for Windows MiKTeX, start the program called MiKTeX Settings (Admin) and click on the button called Refresh FNDB.
10. See section 4.2.1 to generate your local copy of lwarpmk.
11. Once the local version of lwarpmk.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 lwarpmk.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 lwarp before deciding whether to permanently install it.

Just testing!

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

1. At a command line, try executing lwarpmk. If the lwarpmk help message appears, then lwarpmk is already set up. If not, it is easiest to generate and use a local copy. See section 4.2.1.
2. For MiKTeX, try updating the miktex-misc package. This may install the lwarpmk executable for you.

Otherwise, continue with the following:

3. Locate the file lwarpmk.lua, which should be in the scripts directory of the TDS tree. On a T_EX Live or MiKTeX system you may use

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

4. Create lwarpmk:

Unix: Create a symbolic link and make it executable:

- (a) Locate the T_EX Live binaries:

```
Enter ⇒ kpsewhich -var-value TEXMFROOT
```

This will be something like:

```
/usr/local/texlive/<year>
```

The binaries are then located in the bin/<arch> directory under the root:

```
/usr/local/texlive/<year>/bin/<architecture>/
```

In this directory you will find programs such as pdflatex and makeindex.

- (b) In the binaries directory, create a new symbolic link from the binaries directory to lwarpmk.lua:

```
Enter ⇒ ln -s <pathtolwarpmk.lua> lwarpmk
```

- (c) Make the link executable:

```
Enter ⇒ chmod 0755 lwarpmk
```

Windows T_EX Live: Create a new lwarpmk.exe file:

- (a) Locate the T_EX Live binaries as shown above for Unix.
- (b) In the binaries directory, make a *copy* of runscript.exe and call it lwarpmk.exe This will call the copy of lwarpmk.lua which is in the scripts directory of the distribution.

Windows MiKTeX: Create a new lwarpmk.bat file:

- (a) Locate the binaries. These will be in a directory such as:
C:\Program Files\MiKTeX 2.9\miktex\bin\x64
In this directory you will find programs such as pdflatex.exe and makeindex.exe.
 - (b) Create a new file named lwarpmk.bat containing:
texlua "C:\Program Files\MiKTeX 2.9\scripts\lwarp\lwarp.texlua" %*
- This will call the copy of lwarpmk.lua which is in the scripts directory of the distribution.

4.2.1 Using a local copy of lwarpmk

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

1. When compiling the tutorial in section 5, use the lwarpmk option for the lwarp package:
`\usepackage[lwarpmk]{lwarp}`
2. When the tutorial is compiled with pdflatex, the file lwarpmk.lua will be generated along with the other configuration files.
3. lwarpmk.lua may be used for this project:

Unix:

- (a) Make lwarpmk.lua executable:
Enter \Rightarrow `chmod 0755 lwarpmk.lua`
- (b) Compile documents with
Enter \Rightarrow `./lwarpmk.lua html`
Enter \Rightarrow `./lwarpmk.lua print`
etc.
- (c) It may be useful to rename or link to a version without the .lua suffix.

Windows:

Compile documents with either of the following, depending on which command shell is being used:

Enter \Rightarrow `texlua lwarpmk.lua html`
Enter \Rightarrow `texlua lwarpmk.lua print`
etc.

Or:

Enter \Rightarrow `lwarpmk html`
Enter \Rightarrow `lwarpmk 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 45.

```
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 T_EX 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/pdftocrop
```

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

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

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

To install the Poppler utilities to a MacOS machine:

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

To install the Poppler utilities to a Windows machine:

1. See table 2 on page 45.
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`.

File `lwarp_tutorial.txt`

⚠ Note: `.txt` suffix!

⚠ 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, \TeX 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{lwarp_sagebrush.css}

\begin{document}

\maketitle % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

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

\end{document}

```

Table 3: Files created along with the print version

tutorial.pdf:	The PDF output from \LaTeX . The print version of the document.
tutorial_html.tex:	A small .tex file used to create a parallel HTML version of the document, which co-exists with usual the PDF version, and which will have its own auxiliary files. In this way, both PDF and HTML documents may co-exist side-by-side.
Auxiliary files:	The usual \LaTeX files .aux, .log, .out, .toc, .lof, .idx. When an HTML version of the document is created, _html versions of the auxiliary files will also be generated.
lwarpmk.conf:	A configuration file for lwarpmk, which is used to automate the compilation of PDF or HTML versions of the document.
tutorial.lwarpmkconf:	Another configuration file used by lwarpmk, which is only useful if you wish to have several projects residing in the same directory.
.css files:	lwarp.css, lwarp_formal.css, lwarp_sagebrush.css These files are standard for lwarp, and are not meant to be modified by the user.
sample_project.css:	An example of a user-customized css file, which may be used for project-specific changes to the lwarp defaults.
lwarp.xdy:	Used by lwarp while creating an index. This file should not be modified by the user. A custom file may be used instead, if necessary.
lwarp_mathjax.txt:	Inserted into the HTML files when MathJax is used to display math. This file should not be modified by the user.
comment.cut:	A temporary file used by lwarp to conditionally process blocks of text. This file may be ignored.

When the lwarpmk option is given to the lwarp package:

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

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

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

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

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 \Rightarrow `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 \Rightarrow `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 \Rightarrow `lwarpmk again`

Enter \Rightarrow `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 \Rightarrow `lwarpmk printindex`

6. Recompile again to include the index.

Enter \Rightarrow `lwarpmk print`

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

Enter \Rightarrow `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 7.4.8.

⁵Also see section 7.4.9 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 7.4.8.

⁹Also see section 7.4.9 for index options.

5.4 Generating the SVG images

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


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


1. Create the svg images:


```
Enter ⇒ lwarpmk limages
```

```
Enter ⇒ lwarpmk 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 `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 \LaTeX 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.

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 lwarp include:

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

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

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

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

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



Rename it!

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

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

5.7 Customizing the HTML output

⚠ Placement!

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

⚠ Changes!

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

1. Clear all the HTML, PDF, and auxiliary files:

Enter \Rightarrow `lwarpmk clearall`

2. Recompile the print version in order to recreate the configuration files for `lwarpmk`:

Enter \Rightarrow `lwarpmk print`

3. Finally, recompile the HTML version with the new settings:

Enter \Rightarrow `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}`:

Ctrl `tocdepth` **tocdepth:** Sectioning depth of the table of contents. See section 13 for a list of L^AT_EX stack depths.

Ctr SideTOCDepth

sideroc

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

Each subpage of the website has its own small table of contents on the side (the “sideroc”). Its depth is set by SideTOCDepth. This sideroc 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 sideroc 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 sideroc.



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.

Inaccessible 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

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

Named HTML sections, no prefix:

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

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

Named HTML sections, with prefix:

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

```
\usepackage[
  HomeHTMLFilename=mywebsite,
  HTMLFilename={mywebsite-}
]{lwarp}
\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: $\langle filename.css \rangle$ Sets the css file to use for the following files. May be changed before each each sectioning command which would cause a file split.

¹⁰See \SetHTMLFileNumber 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 \LaTeX output. `lwarp_sagebrush.css` is a semi-fancy colored style as shown in this tutorial. Change it to `lwarp_formal.css` for a more formal look, or comment out the `\CSSFilename` command to see the default. `\CSSFilename` may be used before each file break to set the css for individual pagess of the website.

<code>\HTMLLanguage</code>	\HTMLLanguage: The HTML file's <code>html lang</code> tag. Defaults to <code>en-US</code> .
<code>\HTMLAuthor</code>	\HTMLAuthor: The HTML header's meta author. Defaults to <code>\theauthor</code> .
<code>\HTMLDescription</code>	\HTMLDescription: <code>{\langle description \rangle}</code> Sets the HTML <code>description</code> tag for the following files. May be changed before each sectioning command which would cause a file split.
<code>\HTMLFirstPageTop</code>	\HTMLFirstPageTop: <code>{\langle contents \rangle}</code> A user-definable custom action applied to the top of the home page. Useful for logos, etc. Defaults empty. Ignored in print output.
<code>\HTMLPageTop</code>	\HTMLPageTop: <code>{\langle contents \rangle}</code> A user-definable custom action applied to the top of pages other than the home page. Useful for logos, etc. Defaults empty. <code>\LinkHome</code> may be used to place a link back to the homepage. Ignored in print output.
<code>\HTMLPageBottom</code>	\HTMLPageBottom: <code>{\langle contents \rangle}</code> A user-definable custom action applied to the bottom of each web page. Useful for authors, copyright notices, contact information, etc. Defaults empty. <code>\LinkHome</code> 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:

 <code>\tableofcontents</code> 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 <code>warpprint</code>	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 <code>warpprint</code> environment, but unknown packages may cause problems which may be isolated from lwarp using this environment.
----------------------------	--

Env	<code>warpHTML</code>	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.
	<code>\warpprintonly</code>	\warpprintonly: $\{\langle contents \rangle\}$ A macro version of the warpprint environment.
	<code>\warpHTMLonly</code>	\warpHTMLonly: $\{\langle contents \rangle\}$ A macro version of the warpHTML environment.

5.8 Using latexmk

latexmk is a \TeX 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

X_gTeX or LuaTeX may be used instead of E_gTeX.

1. Remove the auxiliary files for the project:

```
Enter ⇒ lwarpmk cleanall
```

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

```
Enter ⇒ xelatex tutorial.tex
```

-or-

```
Enter ⇒ lualatex tutorial.tex
```

When the recompile occurs, the configuration files for lwarpmk are modified to remember which T_EX engine was used. X_gTeX or LuaTeX will be used for future runs of lwarpmk.

3. To recompile the document:

```
Enter ⇒ lwarpmk print
```

-and-

```
Enter ⇒ lwarpmk html
```

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

5.10 Using a glossary

lwarp 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]{lwarp}
```

To process the glossary for the print version:

```
Enter ⇒  lwarpmk printglossary
```

To process the glossary for the HTML version:

```
Enter ⇒  lwarpmk 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_YTeX and LuaTeX 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_YTeX or LuaTeX:

Pkg `fontspec`

- `fontspec` and font choices

ligatures

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

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

- For `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg `fontenc`

- `fontenc`

Pkg `inputenc`

- `inputenc`

Pkg `newunicodechar`

- `newunicodechar`

File	glyphtounicode	<ul style="list-style-type: none"> – \input glyphtounicode.tex – \input glyphtounicode-cmr.tex% from the pdfx package – \pdfgentounicode=1
Pkg	cmap	– cmap
Pkg	textcomp	– textcomp
Pkg	microtype	– microtype is automatically used by lwarp to turn off f,q,t,T,Q ligatures for the same browser-related reasons shown above. Also, the monospaced font is used during HTML tag generation to turn off T _E X ligatures.
	ligatures	

3. \usepackage{lwarp} (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

lwarp 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 T_EX. This option correctly sorts index entries into headings while using Latin languages, but will not work well with others. X_YT_EX or LuaT_EX are recommended for non-Latin languages.

For an older distribution of T_EX, 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 lwarp package loading and options

lwarp supports book, report, and article classes.

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

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 xindy 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 xindy, instead of <code>lwarp.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	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	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.
Opt	OSWindows	

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 \LaTeX print-formatted PDF generation, or to HTML generation.

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

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

Env `warpHTML` Anything which is to be done only for 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` $\{\langle actions \rangle\}$

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

`\warpHTMLonly` $\{\langle actions \rangle\}$

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 lwarp.

`\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` $\{\langle A \text{ description of the web page.} \rangle\}$ 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:

File project.css

File sample_project.css

```
/* ( --- 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.

File lwarp.xdy

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 \TeX 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 `\quad`.

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

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

7.1.3 Text alignment


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

7.1.4 Accents

Native \TeX accents such as `\`` will work, but many more kinds of accents are available when using Unicode-aware \XeTeX and \LuaTeX .

7.1.5 Textcomp

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

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

7.1.6 Superscripts and other non-math uses of math mode

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

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[\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

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

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`, `\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 \LaTeX 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`, `\qqquad`, `\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 warpprint environments. This will nullify the frame for HTML output.
For inline text:	To frame the contents inline with some formatting losses in HTML: This is the default action of <code>\fbox</code> when enclosing a minipage. During HTML output, <code>\fbox</code> nullifies the HTML tags for minipage, <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 minipage and lists:	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 tabular. While an <code>\fbox</code> containing a tabular is valid \LaTeX 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 tabular, or else place the tabular inside a minipage, or use <code>fminipage</code> , described next. Also see below regarding the “Misplaced alignment tab character &” error.
For display tabular, minipages, and lists:	To create a framed minipage in both print and HTML: A new environment <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 tabular. Also see below regarding the “Misplaced alignment tab character &” error.

colored boxes and frames: To create colored frames and boxes: See section 239 for xcolor's `\colorbox` and `\fcolorbox`, and lwarp's additional `\colorboxBlock` and `\fcolorboxBlock`.

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 123 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.

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

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

framed table example The fancybox documentation's example framed table using an `\fbox` containing a

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

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

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


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

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

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

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 `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.

7.3 Cross-references

⚠ empty link

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

7.3.1 Page references

⚠ L^AT_EX page numbers

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

Ex:

```
\ref{sec:name} on page \pageref{sec:name}
in HTML becomes:
“Sec. 1.23 on page (see sec. 1.23)”.
```

`\pagerefPageFor` may be redefined to “page for”, empty, etc. See page 317.

7.3.2 cleveref and varioref packages

⚠ cleveref page numbers

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

Ex:

```
\cpageref{tab:first,tab:second}
in HTML becomes:
“pages for table 4.1 and for table 4.2”
```

See `\cpagerefFor` at page 367 to redefine the message which is printed for page number references.

7.3.3 Hyperlinks, `hyperref`, and `url`

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

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

7.3.4 Footnotes and page notes

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



pfnote numbers

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

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 lwarp.

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

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

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

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

⚠ inaccessible HTML page

`\ForceHTMLTOC` To automatically force the HTML version of the document to have a TOC entry for a starred section, use `\ForceHTMLTOC` just before the `\chapter*` or `\section*`. 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 lwarp supports the native \LaTeX titling commands, and also supports the packages `authblk` and `titling`. If both are used, `authblk` should be loaded before `titling`.

⚠ 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.

7.4.4 tocloft package

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

⚠ tocloft & other packages

7.4.5 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.

7.4.6 pagenote package

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

7.4.7 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
```

7.4.8 glossaries package

`Pkg glossaries` `xindy` is required for glossaries.

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

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

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

Numbered section, on its own HTML page:

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

Unnumbered section, inline with the current HTML page:

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


Unnumbered section, on its own HTML page:

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

Opt	IndexLanguage	The lwarp 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 lwarpmk	printglossary	lwarpmk has the commands lwarpmk printglossary and lwarpmk htmlglossary to process the glossaries created by glossaries using xindy.
Opt lwarpmk	htmlglossary	

7.4.9 Index and the tocibind package

Opt	IndexLanguage	The lwarp 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.
-----	---------------	---

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

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` Use the `tocbibind` `numindex` option to generate a numbered index. Without this option, the index heading has no number.

[numbered index section](#)

See section 62 for `lwarp`'s core index and glossary code, and section 229 for `tocbibind`.

7.5 Math

 **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.

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 \LaTeX into the initial PDF file using the current font¹¹, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the \LaTeX code which generated the math, allowing copy/paste of the \LaTeX math expression into other documents.

SVG image font size 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

¹¹See section 240 regarding fonts and fractions.

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 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 \LaTeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of \LaTeX , and
2. As detokenized printed \LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current \LaTeX values, and the MathJax cross-referencing system is ignored in favor of the \LaTeX internal system, seamlessly integrating with the rest of the \LaTeX code.

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 \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.

<code>lateximage</code>	<ul style="list-style-type: none"> Math appearing inside a <code>lateximage</code>, and therefore also inside a <code>Tikz</code> or <code>picture</code> environment, is rendered as SVG math even if MathJax is used in the rest of the document.
<code>siunitx</code>	<ul style="list-style-type: none"> Usage of <code>siunitx</code> inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use <code>\SI</code> or <code>\si</code> inside <code>\text</code>, where it will be rendered as normal text. <p>https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx</p>
 <code>siunitx</code> inside an equation	
\LaTeX macros	<ul style="list-style-type: none"> 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 337.

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


7.5.2 `siunitx` package

Pkg <code>siunitx</code>	Do not use <code>per-mode=fraction</code> , which cannot be seen by the final <code>pdftotext</code> conversion.
 <code>per-mode</code>	Some units will require that the expression be placed inside math mode.

7.5.3 `units` and `nicefrac` packages



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

7.5.4 `newtxmath` package

Pkg <code>newtxmath</code>	The proper load order is:
 loading sequence	

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

7.6 Graphics

-  **.pdf image files** For `\includegraphics` with .pdf files, the user should provide a .pdf image file, and also a .svg, .png, or .jpg version of the same image. **These should be referred to without a file extension:**
-  **no file extension**



```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```

For print output, lwarp will automatically choose the .pdf if available, other some other format otherwise. For HTML, one of the other formats is used instead.

If a .pdf file is explicitly referred to with its file extension, a link to the .pdf file will appear in the HTML output.

```
\includegraphics{filename.pdf} % creates a link in HTML
```

other image files For .png, .jpg, or .gif image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.


-  **graphics vs. graphicx** If using the older graphics syntax, use both optional arguments for `\includegraphics`. A single optional parameter is interpreted as the newer graphicx syntax. Note that
-  **viewports** viewports are not supported by warp; the entire image will be shown.

units For `\includegraphics`, avoid px and % units for width and height, or enclose them inside warpHTML environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area.


options `\includegraphics` accepts width and height, origin, rotate and scale, plus a new class key.

HTML class With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

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


 **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \TeX , so expect some ugly results for scaling and rotating.

7.6.1 grffile package

 **matching PDF and SVG** `grffile` is supported as-is. File types known to the browser are displayed, and unknown file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

7.6.2 color package

`color` is superseded by `xcolor`, and `lwarp` requires several of the features of `xcolor`.

 **missing colors** It should be sufficient for the user's document to load `color` then load `xcolor` as well.

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 `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for `\colorboxBlock` and `\fcolorboxBlock` are created by `lwarp`'s core if `xcolor` is loaded.

background: none `\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case only the frame is drawn, which can be useful for HTML.

color support Color definitions, models, and mixing are fully supported without any changes required.

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 multirow, longtable, supertabular, or xtab.

Defining environments:

⚠ misplaced alignment
alignment tab character &

- When defining environments or macros which include tabular and instances of the & character, it may be necessary to make & active before the environment or macro is defined, then restore & to its default catcode after, using the following commands. These are 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}
{
  \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.

```
... & \multirow{2}{.5in}{text} & ...
... & \mrowcell & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so `\warpprintonly` and `\warpHTMLonly` must be used to make versions for print and HTML.
- See section 174.2 for `\multicolumnrow`.

⚠ `\multicolumn &`
`\multirow`

lwarp does not support directly combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[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.

⚠ skipped cells

As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines, `\mcolrowcell` must be used for each cell of each column and each row to be skipped:

⚠ empty cells

```
... & \multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

vposn

Note that recent versions of multirow include a new optional vposn argument.

⚠ macro in a table
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

```
\TabularMacro\somemacro & more row contents \\
```

Column specifiers:

⚠ * column specification

- * in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of @ and ! is used at each column, and they are used in that order.

\multirow

- In `\multirow` cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the `\multirow` which do not appear in the HTML version.

⚠ \newcolumn type

- `\newcolumn type` is ignored; unknown column types are set to 1.

Rules:

vertical rules

- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with @ or ! columns, and full-width rules ignore trim.

full-width rules

- \toprule, \midrule, \bottomrule, and \hline ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use \cmidrule followed by \bottomrule, it may be necessary to use:

```
\cmidrule{2-3} \![-2ex]
\bottomrule
```

The optional -2ex is ignored in HTML but improves the visual formatting in the print output.

⚠ \warpprintonly
misplaced \noalign

- For \toprule and \bottomrule, when combined with a warpprint or warpHTML environment, if a “misplaced \noalign” error occurs, change

```
This & That \endhead
```

to

```
\warpprintonly{This & That \endhead}
```

and likewise with the other \end headings. Keep the \endfirsthead row unchanged, as it is still 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 \\\}
\warppHTMLonly{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

⚠ misplaced alignment
alignment tab character &

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

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

See section 7.7.

7.7.3 bigdelim package

⚠ use \mrowcell


\ldelim and \rdelim use \multirow, so \mrowcell must be used in the proper number of empty cells in the same column below \ldelim or \rdelim, but not in cells which are above or below the delimiter:

```
\begin{tabular}{l l l}
<empty> & a & b \\
\ldelim{\}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

$$\text{left} \left\{ \begin{array}{ll} a & b \\ c & d \\ e & f \\ g & h \end{array} \right.$$

7.8 Floats

7.8.1 float, trivfloat, and/or algorithmicx together

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 232.1.


7.8.2 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`.


7.8.3 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.

7.8.4 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. `lwarp` will scale the HTML lengths appropriately.
- Use `warpprint` and `warpHTML` environments to select appropriate values for each case.
- Inside a `warpHTML` environment, manually change `\FBwidth` or `\FBheight` before the `\ffigbox` or `\ttabbox`. Use `\FBwidth` or `\FBheight` normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

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. `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}
```

<p>Len <code>\leftskip</code></p> <p>Len <code>\leftmargini</code></p> <p>Len <code>\TMLvleftskip</code></p> <p>Len <code>\TMLleftmargini</code></p>	<p>These lengths are used by <code>verse</code> and <code>memoir</code> to control the left margin, and they may already be set by the user for print output. New lengths <code>\HTMLvleftskip</code> and <code>\HTMLleftmargini</code> are provided to control the margins in HTML output. These new lengths may be set by the user before any <code>verse</code> environment, and persist until they are manually changed again. One reason to change <code>\HTMLleftmargini</code> is if there is a wide <code>\flagverse</code> in use, such as the word “Chorus”, in which case the value of <code>\HTMLleftmargini</code> should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.</p>
--	--

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

<p>Pkg <code>newclude</code></p> <p> loading</p>	<p><code>newclude</code> modifies <code>\label</code> in a non-adaptive way, so <code>newclude</code> must be loaded before <code>lwarp</code> is loaded.</p>
--	---

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

 **customized spacing** the user's changes do not affect the HTML output.

8 EPUB conversion

lwarp does not produce EPUB documents, but it may be told to modify its HTML output to greatly assist in the conversion. An external program may then be used to finish the conversion to EPUB.

<meta> author To assign the author's name for regular lwarp HTML files, and also for the EPUB, use `\HTMLAuthor {<name>}`. This assigns the name to the `<meta>` author element. It may be set empty, and it defaults to `\theauthor`.

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

Bool `FormatEPUB`

Default false. `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

```
Enter ⇒ lwarpmk cleanall
```

then

```
Enter ⇒ lwarpmk 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.

The resulting files will be ready to be loaded into an EPUB conversion program, such as the open-source program Calibre (<https://calibre-ebook.com/>).

 **search order**

The EPUB conversion program must know what order the files are included. For lwarp projects, set the EPUB conversion software to do a breadth-first search of the files. For Calibre, this option is found in

Preferences → Plugins → File type plugins → HTML to Zip

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

lwarp may be told to modify its HTML output to make it easier to import the HTML document into a word processor. At the time of this writing, it seems that LibreOffice works best at preserving table layout, but it still has some limitations, such as an inability to automatically assign figure and table frames and captions according to user-selected HTML classes. lwarp provides some assistance in locating these frame boundaries, as shown below.

A special boolean is provided to simplify the process of converting lwarp 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 lwarp HTML output for easier import to a word processor, add

```
\booltrue{FormatWordProcessor}
```

to the project's source preamble after lwarp 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. lwarp 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 lwarp

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 `lateximage` 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 lwarp version of a package

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

An `lwarp-<packagename>.sty` package is only used during HTML generation. Its purpose is to pretend to be the original package, while modify anything necessary to create a successful HTML conversion. For many packages it is sufficient to simply provide nullified macros, lengths, counters, etc. for anything which the original package does, while passing the raw text on to be typeset. See the pre-existing `lwarp-` packages for examples.

Anything the user might expect of the original package must be replaced or emulated by the new `lwarp-` package, including package options, user-adjustable counters, lengths, and booleans, and conditional behaviors. In many of these packages, most of the new definitions have a “local” prefix according to the package name, and `@` characters inside the name, which hides these names from the user. In most cases these macros will not need to be emulated for HTML output. Only the “user-facing” macros need to be nullified or emulated.

Each `lwarp-` package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If 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 183 (nttheorem) 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`
File `lwarpmk.lua`

In most installations, `lwarpmk.lua` is an executable file located somewhere the operating system knows about, and it is called by typing “`lwarpmk`” into a terminal.

A project-local copy of `lwarpmk.lua` may be generated, modified, and then used to compile documents:

1. Add the `lwarpmk` option to the `lwarp` package.
2. Recompile the printed version of the document. The `lwarpmk` option causes `lwarp` to create a local copy of `lwarpmk.lua`.
3. The `lwarpmk` option may now be removed from the `lwarp` package.
4. Copy and rename `lwarpmk.lua` to a new file such as `mymake.lua`.
5. Modify `mymake.lua` as desired.
6. If necessary, make `mymake.lua` executable.
7. Use `mymake.lua` instead of `lwarpmk.lua`.

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 `lwarp.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 \TeX 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 `lwarpmk` again followed by `lwarpmk html` or `lwarpmk 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 for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to “for”.

 [empty link](#)

 [cleveref page numbers](#)

Ex:
`\cpageref{tab:first,tab:second}`
 in HTML becomes:
 “pages **for** table 4.1 and **for** table 4.2”
 See `\cpagerefFor` at page 367 to redefine the message which is printed
 for page number references.

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

Use $\text{\texttt{Xe}\TeX}$ or $\text{\texttt{Lua}\TeX}$.

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]{lwarp}` instead of `[warpprint]`.

HTML document contains a single unformatted print document:

- Be sure that the document selects `\usepackage[warpprint]{lwarp}` 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 $\text{\texttt{Xe}\TeX}$ 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 lwarp’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 ~~crude hacks~~ clever tricks. Reference sources are given for many of the solutions, and a quick internet search will provide additional possibilities.

Judgement calls were made, and are often commented. Improvements are possible. The author is open to ideas and suggestions.

Packages were patched for re-use where they provided significant functionality. Examples include xcolor with its color models and conversion to HTML color output, and siunitx which provides many number and unit-formatting options, almost all of which are available in pure-text form, and thus easily used by pdf to text.

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 `lateximage` 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.

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

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.

14 Source Code

This is where the documented source code for lwarp begins, continuing through the following sections all the way to the change log and index at the end of this document.


The following sections document the actual implementation of the lwarp package.

line numbers The small numbers at the left end of a line refer to line numbers in the `lwarp.sty` file.

subjects Blue-colored tags in the left margin aid in quickly identifying the subject of each paragraph.

objects Black-colored tags in the left 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
for PRINT output: the generation of HTML, print, or both forms of output.
for HTML & PRINT:

15 Detecting the T_EX Engine — pdf_latex, lua_latex, xe_latex

```

1 \RequirePackage{iftex}
2
3 \ifLuaTeX
4 \RequirePackage{luatex85}% until the geometry package is updated
5 \fi

```

16 Unicode Input Characters

for HTML & PRINT:

If using pdf_latex, convert a minimal set of Unicode characters. Additional characters may be defined by the user, as needed.

A commonly-used multiply symbol is declared to be \texttimes.

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

```

6
7 \RequirePackage{newunicodechar}
8
9 \newunicodechar{×}{\texttimes}
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 PDF_T_EX, 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 OSWindows.
Prog	Windows	When detected or specified, the operating-system path separator used by lwarp is modified, the boolean usingOSWindows is set true. This boolean may be tested by the user for later use.
Opt	OSWindows	

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

Bool mathjax

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{lwarp}{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{lwarp}{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 `lwarp.xdy`. A customized file should be based on `lwarp.xdy`, and must retain the line

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

```

65 \DeclareStringOption[lwarp.xdy]{xdyFilename}

```

Opt `lwarpmk` Tells `lwarp` to generate a local copy of `lwarpmk` called `lwarpmk.lua`. Useful for archiving for future use. This file may be made executable and acts just like `lwarpmk`.

If `lwarpmk` option, creates a local copy of `lwarpmk.lua`:

```

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 \ifcempty{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` `warppall` Anything in the `warppall` environment will be generated for print or HTML outputs.

```
100 \includecomment{warppall}
```

`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 `lwarpmk`. 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 `lwarp`, and others should only be loaded after.

Packages which should only be loaded before `lwarp` have their own

```
lwarp-<packagename>.sty
```

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

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

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

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

```

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` `{\badpackagename} {\replacementpkgname}`

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{adjmulticol}
144 \LWR@loadafter{addlines}
145 \LWR@loadafter{afterpage}
146 \LWR@loadafter{algorithmicx}
147 \LWR@loadafter{alltt}
148 \LWR@loadafter{amsthm}
149 \LWR@loadafter{anonchap}
150 \LWR@loadafter{anysize}
151 \LWR@loadafter{appendix}
152 \LWR@loadafter{arabicfront}
153 \LWR@loadafter{array}
154 \LWR@loadafter{authblk}
155 \LWR@loadafter{balance}
156 \LWR@loadafter{bigdelim}
157 \LWR@loadafter{bigstrut}
158 \LWR@loadafter{bookmark}

```

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

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

```

259 \LWR@loadafter{sidecap}
260 \LWR@loadafter{sidenotes}
261 \LWR@loadafter{siunitx}
262 \LWR@loadafter{soul}
263 \LWR@loadafter{stabular}
264 \LWR@loadafter{subfig}
265 \LWR@loadafter{supertabular}
266 \LWR@loadafter{tabls}
267 \LWR@loadafter{tabularx}
268 \LWR@loadafter{tabulary}
269 \LWR@loadafter{textarea}
270 % \LWR@loadafter{textcomp}% maybe before lwarp with font packages
271 \LWR@loadafter{textpos}
272 \LWR@loadafter{theorem}
273 \LWR@loadafter{threeparttable}
274 \LWR@loadafter{tikz}
275 \LWR@loadafter{titleps}
276 \LWR@loadafter{titlesec}
277 \LWR@loadafter{titletoc}
278 \LWR@loadafter{titling}
279 \LWR@loadafter{tocbibind}
280 \LWR@loadafter{tocloft}
281 \LWR@loadafter{transparent}
282 \LWR@loadafter{trivfloat}
283 \LWR@loadafter{typearea}
284 \LWR@loadafter{ulem}
285 \LWR@loadafter{upref}
286 \LWR@loadafter{varioref}
287 \LWR@loadafter{verse}
288 \LWR@loadafter{wallpaper}
289 \LWR@loadafter{wrapfig}
290 \LWR@loadafter{xcolor}
291 \LWR@loadafter{xfrac}
292 \LWR@loadafter{xltextra}
293 \LWR@loadafter{xmpinl}
294 \LWR@loadafter{xtab}
295 \LWR@loadafter{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: 296 \begin{warppall}

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

Detects Xe_{La}TeX and Lua_{La}TeX:

```
297 \RequirePackage{iftex}
298 \newif\ifxetexorluatex
299 \ifXeTeX
300     \xetexorluatextrue
301 \else
302     \ifLuaTeX
303         \xetexorluatextrue
304     \else
305         \xetexorluatexfalse
306     \fi
307 \fi

308 \end{warppall}
```

for HTML output: 309 \begin{warppHTML}

```
310 \ifxetexorluatex
311 % ^A \usepackage[no-math]{fontspec}
```

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

```
312 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
313 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
314 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
315 \else
```

pdf_{La}TeX only: Only pre-loaded if pdf_{La}TeX is being used.

Pkg microtype

ligatures Older browsers don't display ligatures. Turn off letter ligatures, keeping \TeX dash and quote ligatures, which may fail on older browsers but at least won't corrupt written words.

```
316 \RequirePackage {microtype}
317
318 \microtypesetup{
319     protrusion=false,
```

```

320     expansion=false,
321     tracking=false,
322     kerning=false,
323     spacing=false}
324
325 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
326 \fi
327 \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:

```

328 \begin{warpHTML}
329 \RequirePackage[paperheight=190in,paperwidth=20in,%
330 left=2in,right=12in,%
331 top=1in,bottom=1in,%
332 ]{geometry}
333 \@twosidefalse
334 \@mparswitchfalse
335 \end{warpHTML}

```

for HTML & PRINT:

```

336 \begin{warpall}

```

Pkg xparse

L^AT_EX3 command argument parsing

```

337 \RequirePackage{xparse}

```

Pkg afterpackage Used to patch titling to add \AddSubtitlePublished.

```

338 \RequirePackage{afterpackage}

```

```

339 \end{warpall}

```

for HTML output:

```

340 \begin{warpHTML}

```

Pkg expl3

L^AT_EX3 programming

```
341 \RequirePackage{expl3}
```

Pkg gettitlestring

Used to emulate \nameref.

```
342 \RequirePackage{gettitlestring}
```

Pkg everyhook

everyhook is used to patch paragraph handling.

```
343 \RequirePackage{everyhook}
344 \end{warpHTML}
```

for HTML & PRINT: 345 \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>

```
346 \RequirePackage{filecontents}
347
348 \@ifpackagelater{filecontents}{2011/10/09}%
349 {}
350 {
351 \newwrite\fcwrite
352 \let\LWR@origfilec@ntents\filec@ntents
353 \def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
354 }
```

```
355 \end{warpall}
```

for HTML output: 356 \begin{warpHTML}

Pkg xifthen

```
357 \RequirePackage{xifthen}
```

Pkg xstring

```
358 \RequirePackage{xstring}

Pkg  xstring

359 \RequirePackage{verbatim}

Pkg  makeidx

360 \RequirePackage{makeidx}
361 \makeindex

Pkg  calc

362 \RequirePackage{calc}

Pkg  refcount

363 \RequirePackage{refcount}

Pkg  newfloat

364 \RequirePackage{newfloat}

Pkg  caption

365 \RequirePackage{caption}

366 \end{warpHTML}

for HTML & PRINT: 367 \begin{warpall}

Pkg  environ

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

368 \RequirePackage{environ}

369 \end{warpall}

for HTML output: 370 \begin{warpHTML}

Pkg  zref

Used for cross-references.

371 \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.

```
372 \PassOptionsToPackage{leqno}{amsmath}
373 \RequirePackage{amsmath}
```

Used to convert lengths for image width/height options.

```
374 \RequirePackage{printlen}

375 \end{warpHTML}
```

22 Loading packages

for HTML output: 376 \begin{warpHTML}

Remember the original \RequirePackage:

```
377 \LetLtxMacro{\LWR@origRequirePackage}{\RequirePackage}
```

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

```
378 \newcommand*\LWR@requirepackagenames{}
```

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

Note that argument 4 is passed directly to \StrBetween.

```
379 \newcommand*\LWR@findword[3][,]{%
380   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
381 }
```

\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.

```
382 \newcommand*\LWR@lookforpackagename[1]{%
```

Find the n'th package name from the list:

```
383 \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.

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

See if the package name was found:

```
385 \IfStrEq{\LWR@strresulttwo}{ }{%
386 { }% no filename
387 { }% yes filename
```

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

```
388 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
389 { % lwarp-* file found
390     \StrSubstitute%
391         {\LWR@requirepackagenames}%
392         {\LWR@strresult}%
393         {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
394 }%
395 { }% no lwarp-* file
396 }% yes filename
397 }
```

`\RequirePackage` [*1: options*] {*2: package names*} [*3: version*]

For each of many package names in a comma-separated list, if an lwarp version of a package exists, select it instead of the \LaTeX version.

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

Redirect up to nine names:

```
399 \renewcommand*{\LWR@requirepackagenames}{#2}
400 \LWR@lookforpackagename{1}
401 \LWR@lookforpackagename{2}
402 \LWR@lookforpackagename{3}
403 \LWR@lookforpackagename{4}
404 \LWR@lookforpackagename{5}
405 \LWR@lookforpackagename{6}
406 \LWR@lookforpackagename{7}
407 \LWR@lookforpackagename{8}
408 \LWR@lookforpackagename{9}
```

`\RequirePackage` depending on the options and version:

```
409 \IfValueTF{#1}
410 { % options given
```

```

411 \IfValueTF{#3}% version given?
412 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
413 {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
414 }
415 {% no options given
416 \IfValueTF{#3}% version given?
417 {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
418 {\LWR@origRequirePackage{\LWR@requirepackagenames}}
419 }
420 }
421 \LetLtxMacro{\usepackage}{\RequirePackage}

```

`\LWR@ProvidesPackagePass` `{\langle pkgname \rangle} [\langle version \rangle]`

Uses the original package, including options.

```

422 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
423 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
424 \IfValueTF{#2}
425 {\ProvidesPackage{lwarp-#1}[#2]}
426 {\ProvidesPackage{lwarp-#1}}
427 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
428 \ProcessOptions\relax
429
430 \IfValueTF{#2}
431 {\LWR@origRequirePackage{#1}[#2]}
432 {\LWR@origRequirePackage{#1}}
433 }

```

`\LWR@ProvidesPackageDrop` `{\langle pkgname \rangle} [\langle version \rangle]`

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

```

434 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
435 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
436 \IfValueTF{#2}
437 {\ProvidesPackage{lwarp-#1}[#2]}
438 {\ProvidesPackage{lwarp-#1}}
439 \DeclareOption*{}
440 \ProcessOptions\relax
441 }

442 \end{warpHTML}

```

23 File handles

Defines file handles for writes.

for HTML & PRINT: 443 \begin{warpall}

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

444 \newwrite\LWR@quickfile%

445 \end{warpall}

for HTML output: 446 \begin{warpHTML}

\LWR@lateximagesfile For lateximages.txt.

447 \newwrite\LWR@lateximagesfile

448 \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 \LaTeX .

for HTML output: 449 \begin{warpHTML}

\include {<filename>}

\@include {<filename>} Modified to load _html_inc files.

450 \def\@include#1 {%

451 \immediate\openout\LWR@quickfile #1_html_inc.tex% new

452 \immediate\write\LWR@quickfile{\string\input{#1.tex}}% new

```

453 \immediate\closeout\LWR@quickfile% new
454 \LWR@origclearpage% \changed
455 \if@filesw
456   \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
457 \fi
458 \@tempwattrue
459 \if@partsw
460   \@tempwafalse
461   \edef\reserved@a{\@partlist\do
462     \ifx\reserved@a\reserved@b\@tempwattrue\fi}%
464 \fi
465 \if@tempswa
466   \let\@auxout\@partaux
467   \if@filesw
468     \immediate\openout\@partaux #1_html_inc.aux % changed
469     \immediate\write\@partaux{\relax}%
470   \fi
471   \@input{#1_html_inc.tex}% changed
472   \LWR@origclearpage% changed
473   \@writeckpt{#1}%
474   \if@filesw
475     \immediate\closeout\@partaux
476   \fi
477 \else
478   \deadcycles\z@
479   \@nameuse{cp@#1}%
480 \fi
481 \let\@auxout\@mainaux%
482 }

483 \end{warpHTML}

```

25 Copying a file

for HTML output: 484 \begin{warpHTML}

\LWR@copyfile {<source filename>} {<destination filename>}

Used to copy the .toc file to .sidetoc to re-print the toc in the sideroc navigation pane.

```

485 \newwrite\LWR@copyoutfile % open the file to write to
486 \newread\LWR@copyinfile % open the file to read from
487
488 \newcommand*\LWR@copyfile[2]{%

```

```

489 \immediate\openout\LWR@copyoutfile=#2
490 \openin\LWR@copyinfile=#1
491 \begingroup\endlinechar=-1
492 \makeatletter
493 \loop\unless\ifeof\LWR@copyinfile
494   \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
495 %   \LWR@fileline\par % print the content into the pdf
496 % print the content:
497   \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
498 \repeat
499 \closeout\LWR@copyoutfile
500 \endgroup
501 }

502 \end{warpHTML}

```

26 Debugging messages

```
503 \begin{warpall}
```

Bool LWR@tracinglwarp True if tracing is turned on.

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

\tracinglwarp Turns on the debug tracing messages.

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

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

```

506 \newcommand{\LWR@traceinfo}[1]{%
507 \ifbool{LWR@tracinglwarp}%
508 {%
509   \typeout{*** lwarp: #1}%
510   % \PackageInfo{lwarp}{#1 : }%
511 }%
512 {}%
513 }

```

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

```

514 \newbool{HTMLDebugComments}
515 \boolfalse{HTMLDebugComments}

```

If `\tracinglwarp`, show where preamble hooks occur:

```
516 \AfterEndPreamble{
517 \LWR@traceinfo{AfterEndPreamble}
518 }
519
520 \AtBeginDocument{
521 \LWR@traceinfo{AtBeginDocument}
522 }
```

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.

```
523 \newbool{FormatEPUB}
524 \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.

```
525 \newbool{FormatWordProcessor}
526 \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

```
527 \newbool{HTMLMarkFloats}
528 \booltrue{HTMLMarkFloats}
```

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

```
529 \end{warpall}
```

28 Remembering original formatting macros

for HTML output: 530 \begin{warpHTML}

Remember original definitions of formatting commands. Will be changed to HTML commands for most uses. Will be temporarily restored to original meaning inside any lateximage environment. Also nullify unused commands.

```
531 \LetLtxMacro{\LWR@origtextrm}{\textrm}
532 \LetLtxMacro{\LWR@origtextsf}{\textsf}
533 \LetLtxMacro{\LWR@origtexttt}{\texttt}
534 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
535 \LetLtxMacro{\LWR@origtextbf}{\textbf}
536 \LetLtxMacro{\LWR@origtextmd}{\textmd}
537 \LetLtxMacro{\LWR@origtextit}{\textit}
538 \LetLtxMacro{\LWR@origtextsl}{\textsl}
539 \LetLtxMacro{\LWR@origtextsc}{\textsc}
540 \LetLtxMacro{\LWR@origtextup}{\textup}
541 \LetLtxMacro{\LWR@origemph}{\emph}
542
543 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
544 \LetLtxMacro{\LWR@origsfffamily}{\sffamily}
545 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
546 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
547 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
548 \LetLtxMacro{\LWR@origupshape}{\upshape}
549 \LetLtxMacro{\LWR@origslshape}{\slshape}
550 \LetLtxMacro{\LWR@origscshape}{\scshape}
551 \LetLtxMacro{\LWR@origitshape}{\itshape}
552 \LetLtxMacro{\LWR@origem}{\em}
553 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
554
555 \let\LWR@origraggedright\raggedright
556 \let\LWR@origonecolumn\onecolumn
557
558 \let\LWR@origsp\sp
559 \let\LWR@origsb\sb
560 \let\LWR@origtextsuperscript\textsuperscript
561 \let\LWR@origtextsubscript\textsubscript
562 \LetLtxMacro\LWR@origunderline\underline
563
564 \let\LWR@origscriptsize\scriptsize
565
566 \let\LWR@orignewpage\newpage
```

```

567
568 \let\LWR@origpagestyle\pagestyle
569 \let\LWR@origthispagestyle\thispagestyle
570 \let\LWR@origpagenumbering\pagenumbering
571
572 \LetLtxMacro{\LWR@origminipage}{\minipage}
573 \let\LWR@origendminipage\endminipage
574 \LetLtxMacro{\LWR@origparbox}{\parbox}
575
576 \let\LWR@orignewline\newline
577
578 \LetLtxMacro{\LWR@origitem}{\item}
579
580 \LetLtxMacro{\LWR@origitemize}{\itemize}
581 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
582 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
583 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
584 \LetLtxMacro{\LWR@origdescription}{\description}
585 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
586
587 \let\LWR@origpar\par
588
589 \LetLtxMacro{\LWR@origfootnote}{\footnote}
590 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
591
592 \let\LWR@origclearpage\clearpage
593 \let\clearpage\relax
594 \let\cleardoublepage\relax
595 \end{warpHTML}

```

29 Accents

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

for HTML output: 596 \begin{warpHTML}

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

```
597 \AtBeginDocument{
```

The following are restored for print when inside a `lateximage`.

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


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

For pdf_{La}TeX, additional work is required:

```
599 \ifPDFTeX
600 \LetLtxMacro{\LWR@origequalaccent}{\=}
601 \LetLtxMacro{\LWR@origdotaccent}{\cdot}
602 \LetLtxMacro{\LWR@origu}{\u}
603 \LetLtxMacro{\LWR@origv}{\v}
604 \LetLtxMacro{\LWR@origc}{\c}
605 \LetLtxMacro{\LWR@origd}{\d}
606 \LetLtxMacro{\LWR@origb}{\b}
```

The HTML redefinitions follow.

For pdf_{La}TeX, Unicode diacritical marks are used:

```
607 \renewcommand*{\=}[1]{\#1\HTMLUnicode{0305}}
608 \renewcommand*{\cdot}[1]{\#1\HTMLUnicode{0307}}
609 \renewcommand*{\u}[1]{\#1\HTMLUnicode{0306}}
610 \renewcommand*{\v}[1]{\#1\HTMLUnicode{030C}}
611 \renewcommand*{\c}[1]{\#1\HTMLUnicode{0327}}
612 \renewcommand*{\d}[1]{\#1\HTMLUnicode{0323}}
613 \renewcommand*{\b}[1]{\#1\HTMLUnicode{0331}}
614 \fi
```

For all engines, a Unicode diacritical tie is used:

```
615 \def\LWR@t#1#2{\#1\HTMLUnicode{0361}\#2}
616 \renewcommand*{\t}[1]{\LWR@t#1}
```

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

```
617 \ifPDFTeX
618 \newcommand*{\LWR@restoreorigaccents}{%
619 \LetLtxMacro{\=}{\LWR@origequalaccent}%
620 \LetLtxMacro{\cdot}{\LWR@origdotaccent}%
621 \LetLtxMacro{\u}{\LWR@origu}%
622 \LetLtxMacro{\v}{\LWR@origv}%
623 \LetLtxMacro{\t}{\LWR@origt}%
624 \LetLtxMacro{\c}{\LWR@origc}%
625 \LetLtxMacro{\d}{\LWR@origd}%
626 \LetLtxMacro{\b}{\LWR@origb}%
627 }
628 \else% XeLaTeX, LuaLaTeX:
629 \newcommand*{\LWR@restoreorigaccents}{%
630 \LetLtxMacro{\t}{\LWR@origt}%
631 }
632 \fi
```

```
633 }% AtBeginDocument
```

```
634 \end{warpHTML}
```

30 Configuration Files

```
635 \begin{warpprint}
636 \typeout{lwarp: generating configuration files}
637 \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.

```
638 \begin{warpprint}
639 \immediate\openout\LWR@quickfile=\jobname_html.tex
640 \immediate\write\LWR@quickfile{%
641 \detokenize{\PassOptionsToPackage}%
642 {warpHTML,BaseJobname=\jobname}{lwarp}%
643 }
644 \immediate\write\LWR@quickfile{%
645 \detokenize{\input}\string{\jobname.tex}\string }%
646 }
647 \immediate\closeout\LWR@quickfile
648 \end{warpprint}
```

30.2 lwarpmk.conf

File `lwarpmk.conf` `lwarpmk.conf` is automatically (re-)created by the `lwarp` package when executing `pdflatex <project.tex>`, or similar for `xelatex` or `lualatex`, in print-document generation mode, which is the default unless the `warpHTML` option is given. `lwarpmk.conf` is then used by the utility `lwarpmk`.

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: 649 \begin{warpprint}
650 \ifcsdef{LWR@quickfile}{-}{\newwrite{LWR@quickfile}}
651 \immediate\openout\LWR@quickfile=lwarpmk.conf
652 \ifbool{usingOSWindows}{
653 \immediate\write\LWR@quickfile{opsystem = "Windows"}
654 }{
655 \immediate\write\LWR@quickfile{opsystem = "Unix"}
656 }
657 \ifPDFTeX
658 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
659 \fi
660 \ifXeTeX
661 \immediate\write\LWR@quickfile{latexname = "xelatex"}
662 \fi
663 \ifLuaTeX
664 \immediate\write\LWR@quickfile{latexname = "lualatex"}
665 \fi
666 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
667 \immediate\write\LWR@quickfile{%
668 homehtmlfilename = "\HomeHTMLFilename"%
669 }
670 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
671 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}}"}
672 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
673 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
674 \immediate\closeout\LWR@quickfile
675 \end{warpprint}

```

30.3 project.lwarpmkconf

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

```

676 \begin{warpprint}
677 \ifcsdef{LWR@quickfile}{-}{\newwrite{LWR@quickfile}}
678 \immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
679 \ifbool{usingOSWindows}{
680 \immediate\write\LWR@quickfile{opsystem = "Windows"}
681 }{
682 \immediate\write\LWR@quickfile{opsystem = "Unix"}
683 }
684 \ifPDFTeX
685 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
686 \fi
687 \ifXeTeX
688 \immediate\write\LWR@quickfile{latexname = "xelatex"}

```

```

689 \fi
690 \ifLuaTeX
691 \immediate\write\LWR@quickfile{latexname = "lua\latex"}
692 \fi
693 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
694 \immediate\write\LWR@quickfile{%
695 homehtmlfilename = "\HomeHTMLFilename"%
696 }
697 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
698 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
699 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
700 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
701 \immediate\closeout\LWR@quickfile
702 \end{warpprint}

```

30.4 lwarp.css

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

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

```

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

```

```
728     text-indent: -.5em ; }
729 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
730     text-indent: -.5em ; }
731 a.tocparagraph {display: inline-block ; margin-left: 4em ;
732     text-indent: -.5em ; }
733 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
734     text-indent: -.5em ; }
735 a.tocfigure {margin-left: 0em}
736 a.tocsubfigure {margin-left: 2em}
737 a.toc table {margin-left: 0em}
738 a.tocsubtable {margin-left: 2em}
739 a.toc theorem {margin-left: 0em}
740 a.toc lstlisting {margin-left: 0em}
741
742
743 body {
744     font-family: "DejaVu Serif", "Bitstream Vera Serif",
745         "Lucida Bright", Georgia, serif;
746     background: #FAF7F4 ;
747     color: black ;
748     margin: 0em ;
749     padding: 0em ;
750     font-size: 100% ;
751     line-height: 1.2 ;
752 }
753
754 p {margin: 1.5ex 0em 1.5ex 0em ;}
755
756 /* Holds a section number to add space between it and the name */
757 span.sectionnumber { margin-right: 0em }
758
759 /* Inserted in front of index lines */
760 span.indexitem {margin-left: 0em}
761 span.indexsubitem {margin-left: 2em}
762 span.indexsubsubitem {margin-left: 4em}
763
764 div.hidden, span.hidden { display: none ; }
765
766 kbd {
767     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
768         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
769         "Courier New", monospace;
770     font-size: 100% ;
771 }
772
773 pre { padding: 3pt ; }
774
775 span.strong { font-weight: bold; }
776
777 span.textmd { font-weight: normal; }
```

```
778
779 span.textsc { font-variant: small-caps; }
780
781 span.textsl { font-style: oblique; }
782
783 span.textup { font-variant: normal; }
784
785 span.textrm {
786     font-family: "DejaVu Serif", "Bitstream Vera Serif",
787     "Lucida Bright", Georgia, serif;
788 }
789
790 span.textsf {
791     font-family: "DejaVu Sans", "Bitstream Vera Sans",
792     Geneva, Verdana, sans-serif ;
793 }
794
795 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
796
797 span.underline {
798     text-decoration: underline ;
799     text-decoration-skip ;
800 }
801
802
803 /* For realscripts */
804 .supsubscript {
805     display: inline-block;
806     text-align: left ;
807 }
808
809 .supsubscript sup,
810 .supsubscript sub {
811     position: relative;
812     display: block;
813     font-size: .5em;
814     line-height: 1;
815 }
816
817 .supsubscript sup {
818     top: .5em;
819 }
820
821 .supsubscript sub {
822     top: .5em;
823 }
824
825 span.attribution {
826     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
827 }
```

```
828
829 span.citetitle {
830   margin-left: 1em ; font-size: 80% ; font-style: oblique;
831 }
832
833 span.poemtitle {
834   font-size: 120% ; font-weight: bold;
835 }
836
837 blockquote {
838   margin-left: 0px ;
839   margin-right: 0px ;
840 }
841
842 /* quotchap is for the quotchap package */
843 div.quotchap {
844   font-style: oblique ;
845   overflow-x: auto ;
846   margin-left: 2em ;
847   margin-right: 2em ;
848 }
849
850 blockquote p, div.quotchap p {
851   line-height: 1.5;
852   text-align: left ;
853   font-size: .85em ;
854   margin-left: 3em ;
855   margin-right: 3em ;
856 }
857
858 /* qauthor is for the quotchap package */
859 div.qauthor {
860   display: block ;
861   text-align: right ;
862   margin-left: auto ;
863   margin-right: 2em ;
864   font-size: 80% ;
865   font-variant: small-caps;
866 }
867
868 div.qauthor p {
869   text-align: right ;
870 }
871
872 blockquotation {
873   margin-left: 0px ;
874   margin-right: 0px ;
875 }
876
877 blockquotation p {
```

```
878 line-height: 1.5;
879 text-align: left ;
880 font-size: .85em ;
881 margin-left: 3em ;
882 margin-right: 3em ;
883 }
884
885 div.epigraph {
886 line-height: 1.2;
887 text-align: left ;
888 padding: 3ex 1em 0ex 1em ;
889 /* margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
890 margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
891 /* margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
892 font-size: .85em ;
893 max-width: 27em ;
894 }
895
896
897
898 div.epigraphsource{
899 text-align:right ;
900 margin-left:auto ;
901 /* max-width: 50% ; */
902 border-top: 1px solid #A0A0A0 ;
903 padding-bottom: 3ex ;
904 line-height: 1.2;
905 }
906
907 div.epigraph p { padding: .5ex ; margin: 0ex ;}
908 div.epigraphsource p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
909
910
911 /* lettrine package: */
912 span.lettrine { font-size: 3ex ; float: left ; }
913 span.lettrinetext { font-variant: small-caps ; }
914
915 /* ulem and soul packages: */
916 span.uline {
917 text-decoration: underline ;
918 text-decoration-skip ;
919 }
920
921 span.uuline {
922 text-decoration: underline ;
923 text-decoration-skip ;
924 text-decoration-style: double ;
925 }
926
927 span.uwave {
```



```
928     text-decoration: underline ;
929     text-decoration-skip ;
930     text-decoration-style: wavy ;
931 }
932
933 span.sout {
934     text-decoration: line-through ;
935 }
936
937 span.xout {
938     text-decoration: line-through ;
939 }
940
941 span.dashuline {
942     text-decoration: underline ;
943     text-decoration-skip ;
944     text-decoration-style: dashed ;
945 }
946
947 span.dotuline {
948     text-decoration: underline ;
949     text-decoration-skip ;
950     text-decoration-style: dotted ;
951 }
952
953 span.letterspacing { letter-spacing: .2ex ; }
954
955 span.capsspacing {
956     font-variant: small-caps ;
957     letter-spacing: .1ex ;
958 }
959
960 span.highlight { background: #F8E800 ; }
961
962
963
964
965 html body {
966     margin: 0 ;
967     line-height: 1.2;
968 }
969
970
971 body div {
972     margin: 0ex;
973 }
974
975
976 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
977 {
```

```
978     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
979           "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
980           "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
981           "Times New Roman", serif;
982     font-style: normal ;
983     font-weight: bold ;
984     text-align: left ;
985 }
986
987 h1 { /* title of the entire website, used on each page */
988     text-align: center ;
989     font-size: 2.5em ;
990     padding: .4ex 0em 0em 0em ;
991 }
992 h2 { font-size: 2.25em }
993 h3 { font-size: 2em }
994 h4 { font-size: 1.75em }
995 h5 { font-size: 1.5em }
996 h6 { font-size: 1.25em }
997 span.paragraph {font-size: 1em ; font-variant: normal ;
998     margin-right: 1em ; }
999 span.subparagraph {font-size: 1em ; font-variant: normal ;
1000     margin-right: 1em ; }
1001
1002
1003
1004 /* Title of the file */
1005 h1 {
1006     margin: 0em 0em 0em 0em ;
1007     line-height: 1.3;
1008     text-align: center ;
1009 }
1010
1011 /* Part */
1012 h2 {
1013     margin: 1ex 0em 1ex 0em ;
1014     line-height: 1.3;
1015     text-align: center ;
1016 }
1017
1018 /* Chapter */
1019 h3 {
1020     margin: 3ex 0em 1ex 0em ;
1021     line-height: 1.3;
1022 }
1023
1024 /* Section */
1025 h4 {
1026     margin: 3ex 0em 1ex 0em ;
1027     line-height: 1.3;
```

```
1028 }
1029
1030 /* Sub-Section */
1031 h5 {
1032     margin: 3ex 0em 1ex 0em ;
1033     line-height: 1.3;
1034 }
1035
1036 /* Sub-Sub-Section */
1037 h6 {
1038     margin: 3ex 0em 1ex 0em ;
1039     line-height: 1.3;
1040 }
1041
1042
1043 div.titlepage {
1044     text-align: center ;
1045 }
1046
1047 .footnotes {
1048     font-size: .85em ;
1049     margin: 3ex 1em 0ex 1em ;
1050     padding-bottom: 1ex ;
1051     border-top: 1px solid silver ;
1052 }
1053
1054 .marginpar {
1055     max-width:50%;
1056     float:right;
1057     text-align:left;
1058     margin: 1ex 0.5em 1ex 1em ;
1059     padding: 1ex 0.5em 1ex 0.5em ;
1060     font-size: 85% ;
1061     border-top: 1px solid silver ;
1062     border-bottom: 1px solid silver ;
1063     overflow-x: auto;
1064 }
1065
1066 .marginpar br { margin-bottom: 2ex ; }
1067
1068 div.marginblock {
1069     max-width:50%;
1070     float:right;
1071     text-align:left;
1072     margin: 1ex 0.5em 1ex 1em ;
1073     padding: 1ex 0.5em 1ex 0.5em ;
1074     overflow-x: auto;
1075 }
1076
1077 div.marginblock div.minipage {
```

```
1078     display: block ;
1079     margin: 0pt auto 0pt auto ;
1080 }
1081
1082 div.marginblock div.minipage p { font-size: 85%}
1083
1084 div.marginblock br { margin-bottom: 2ex ; }
1085
1086
1087 section.textbody div.footnotes{
1088     margin: 3ex 0em 0ex 0em ;
1089     border-bottom: 2px solid silver ;
1090 }
1091
1092 .footnoteheader {
1093     border-top: 2px solid silver ;
1094     margin-top: 3ex ;
1095     padding-top: 1ex ;
1096     font-weight: bold ;
1097 }
1098
1099 .mpfootnotes {
1100     text-align: left ;
1101     font-size: .85em ;
1102     margin-left: 1em ;
1103     border-top: 1px solid silver ;
1104 }
1105
1106 /* Remove footnote top border in the title page. */
1107 div.titlepage div.mpfootnotes {
1108     border-top: none ;
1109 }
1110
1111
1112
1113 ol {
1114     margin: 1ex 1em 1ex 0em;
1115     line-height: 1.2;
1116 }
1117
1118 ul, body dir, body menu {
1119     margin: 1ex 1em 1ex 0em;
1120     line-height: 1.2;
1121 }
1122
1123 li { margin: 0ex 0em 1ex 0em; }
1124
1125 html {
1126     margin: 0;
1127     padding: 0;
```

```
1128 }
1129
1130 .programlisting {
1131     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1132         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1133         "Courier New", monospace;
1134     margin: 1ex 0ex 1ex 0ex ;
1135     padding: .5ex 0pt .5ex 0pt ;
1136     overflow-x: auto;
1137 }
1138
1139 section.textbody>pre.programlisting {
1140 border-top: 1px solid silver ;
1141 border-bottom: 1px solid silver ;
1142 }
1143
1144
1145 .inlineprogramlisting {
1146     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1147         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1148         "Courier New", monospace;
1149     overflow-x: auto;
1150 }
1151
1152 span.listinglabel {
1153     display: inline-block ;
1154     font-size: 70% ;
1155     width: 4em ;
1156     text-align: right ;
1157     margin-right: 2em ;
1158 }
1159
1160 div.abstract {
1161     margin: 2em 5% 2em 5% ;
1162     padding: 1ex 1em 1ex 1em ;
1163     /* font-weight: bold ; */
1164     font-size: 90% ;
1165     text-align: left ;
1166 }
1167
1168 div.abstract dl {line-height:1.5;}
1169 div.abstract dt {color:#304070;}
1170
1171 div.abstracttitle{
1172     font-family: "URW Classico", Optima, "Linux Biolinum O",
1173         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1174         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1175     font-weight:bold;
1176     font-size:1.25em;
1177     text-align: center ;
```

```
1178 }
1179
1180 span.abstractrunintitle{
1181     font-family: "URW Classico", Optima, "Linux Biolinum O",
1182         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1183         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1184     font-weight:bold;
1185 }
1186
1187
1188 .verbatim {
1189     overflow-x: auto ;
1190 }
1191
1192 .alltt {
1193     overflow-x: auto ;
1194 }
1195
1196
1197 .bverbatim {
1198     margin: 1ex Opt 1ex Opt ;
1199     padding: .5ex Opt .5ex Opt ;
1200     overflow-x: auto ;
1201 }
1202
1203 .lverbatim {
1204     margin: 1ex Opt 1ex Opt ;
1205     padding: .5ex Opt .5ex Opt ;
1206     overflow-x: auto ;
1207 }
1208
1209 .fancyvrb {
1210     font-size:.85em ;
1211     margin: 3ex Opt 3ex Opt
1212 }
1213
1214 .fancyvrblabel {
1215     font-weight:bold;
1216     text-align: center ;
1217 }
1218
1219
1220 .verse {
1221     font-family: "Linux Libertine Mono O", "Lucida Console",
1222         "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1223         "Liberation Mono", "FreeMono", "Andale Mono",
1224         "Nimbus Mono L", "Courier New", monospace;
1225     margin-left: 1em ;
1226 }
1227
```

```
1228
1229 div.singlespace { line-height: 1.2 ; }
1230 div.onehalfspace { line-height: 1.5 ; }
1231 div.doublespace { line-height: 2 ; }
1232
1233
1234
1235
1236
1237 /* Minipage environments, vertically aligned to top, center, bottom: */
1238 .minipage, .fminipage, .fcolorminipage {
1239     /* display: inline-block ; */
1240     /* Mini pages which follow each other will be tiled. */
1241     margin: .25em .25em .25em .25em;
1242     padding: .25em .25em .25em .25em;
1243     display: inline-flex;
1244     flex-direction: column ;
1245     overflow: auto;
1246 }
1247
1248 /* Paragraphs in the flexbox did not collapse their margins. */
1249 /* Have not yet researched this. */
1250 .minipage p {margin: .75ex 0em .75ex 0em ;}
1251
1252 .fbboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1253 .fcolorbox .minipage, .fcolorboxBlock .minipage
1254     {border: none ; background: none;}
1255
1256 .fbbox, .fbboxBlock { border: 1px solid black ; }
1257
1258 .fbbox, .fbboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1259 .fminipage, .fcolorminipage
1260     {display: inline-block}
1261
1262 .shadowbox, .shabox {
1263     border: 1px solid black;
1264     box-shadow: 3px 3px 3px #808080 ;
1265     border-radius: 0px ;
1266     padding: .4ex .3em .4ex .3em ;
1267     margin: 0pt .3ex 0pt .3ex ;
1268     display: inline-block ;
1269 }
1270
1271 .doublebox {
1272     border: 3px double black;
1273     border-radius: 0px ;
1274     padding: .4ex .3em .4ex .3em ;
1275     margin: 0pt .3ex 0pt .3ex ;
1276     display: inline-block ;
1277 }
```

```
1278
1279 .ovalbox, .Ovalbox {
1280     border: 1px solid black;
1281     border-radius: 1ex ;
1282     padding: .4ex .3em .4ex .3em ;
1283     margin: 0pt .3ex 0pt .3ex ;
1284     display: inline-block ;
1285 }
1286
1287 .Ovalbox { border-width: 2px ; }
1288
1289 .framebox {
1290     border: 1px solid black;
1291     border-radius: 0px ;
1292     padding: .3ex .2em 0ex .2em ;
1293     margin: 0pt .1ex 0pt .1ex ;
1294     display: inline-block ;
1295 }
1296
1297
1298 .mdframed {
1299 /*     padding: 0ex ; */
1300 /*     border: 1px solid black; */
1301 /*     border-radius: 0px ; */
1302     padding: 0ex ;
1303     margin: 3ex 5% 3ex 5% ;
1304 /*     display: inline-block ; */
1305 }
1306
1307 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1308
1309 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1310
1311 .mdframedtitle {
1312     padding: .5em ;
1313     display: block ;
1314     font-size: 130%
1315 }
1316
1317 .mdframedsubtitle {
1318     padding: 0ex .5em 0ex .5em ;
1319     display: block ;
1320     font-size: 115% ;
1321 }
1322
1323 .mdframedsubsubtitle {
1324     padding: 0ex .5em 0ex .5em ;
1325     display: block ;
1326 }
1327
```



```
1328 .mdtheorem {
1329     padding: 0ex .5em 0ex .5em ;
1330     margin: 3ex 5% 3ex 5% ;
1331 /*     display: inline-block ; */
1332 }
1333
1334
1335 /* framed package */
1336 .framed, pre.boxedverbatim, fcolorbox {
1337     margin: 3ex 0em 3ex 0em ;
1338     border: 1px solid black;
1339     border-radius: 0px ;
1340     padding: .3ex 1em 0ex 1em ;
1341     display: block ;
1342 }
1343
1344 .snugframed {
1345     margin: 3ex 0em 3ex 0em ;
1346     border: 1px solid black;
1347     border-radius: 0px ;
1348     display: block ;
1349 }
1350
1351 .framedleftbar {
1352     margin: 3ex 0em 3ex 0em ;
1353     border-left: 3pt solid black;
1354     border-radius: 0px ;
1355     padding: .3ex .2em .3ex 1em ;
1356     display: block ;
1357 }
1358
1359 .framedtitle {
1360     margin: 0em ;
1361     padding: 0em ;
1362     font-size: 130%
1363 }
1364
1365 .framedtitle p { padding: .3em }
1366
1367
1368
1369 dl {
1370     margin: 1ex 2em 1ex 0em;
1371     line-height: 1.3;
1372 }
1373
1374 dl dt {
1375     margin-top: 1ex;
1376     margin-left: 1em ;
1377     font-weight: bold;
```

```
1378 }
1379
1380 dl dd p { margin-top: 0em; }
1381
1382
1383 nav.toc, nav.lof, nav.lot, nav.lol, nav.lothm {
1384     font-family: "URW Classico", Optima, "Linux Biolinum O",
1385         "DejaVu Sans", "Bitstream Vera Sans",
1386         Geneva, Verdana, sans-serif ;
1387     margin-bottom: 4ex ;
1388 }
1389
1390 nav.toc p, nav.lof p, nav.lot p, nav.lol p, nav.lothm p {
1391     line-height: 1.2 ;
1392     margin-top: .5ex ;
1393     margin-bottom: .5ex ;
1394     font-size: .9em ;
1395 }
1396
1397
1398
1399 img, img.hyperimage, img.borderimage {
1400     max-width: 600px ;
1401     border: 1px solid silver ;
1402     box-shadow: 3px 3px 3px #808080 ;
1403     padding: .5% ;
1404     margin: .5% ;
1405     background: none ;
1406 }
1407
1408 img.inlineimage{
1409     padding: 0px ;
1410     box-shadow: none ;
1411     border: none ;
1412     background: none ;
1413     margin: 0px ;
1414     display: inline-block ;
1415     border-radius: 0px ;
1416 }
1417
1418 img.logoimage{
1419     max-width: 300px ;
1420     box-shadow: 3px 3px 3px #808080 ;
1421     border: 1px solid black ;
1422     background: none ;
1423     padding: 0 ;
1424     margin: .5ex ;
1425     border-radius: 10px ;
1426 }
1427
```

```
1428
1429 .section {
1430 /*
1431     To have each section float relative to each other:
1432 */
1433 /*
1434     display: block ;
1435     float: left ;
1436     position: relative ;
1437     background: white ;
1438     border: 1px solid silver ;
1439     padding: .5em ;
1440 */
1441     margin: 0ex .5em 0ex .5em ;
1442     padding: 0 ;
1443 }
1444
1445
1446 figure {
1447     margin: 3ex auto 3ex auto ;
1448     padding: 1ex 1em 1ex 1em ;
1449     overflow-x: auto ;
1450 }
1451
1452
1453 /* To automatically center images in figures: */
1454 /*
1455 figure img.inlineimage {
1456     margin: 0ex auto 0ex auto ;
1457     display: block ;
1458 }
1459 */
1460
1461 /* To automatically center minipages in figures: */
1462 /*
1463 figure div.minipage, figure div.minipage div.minipage {
1464     margin: 1ex auto 1ex auto ;
1465     display: block ;
1466 }
1467 */
1468
1469 figure div.minipage p { font-size: 85% ; }
1470
1471 figure.subfigure, figure.subtable {
1472     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1473 }
1474
1475 figcaption .minipage { margin: 0 ; padding: 0 }
1476
1477 div.minipage figure { border: none ; box-shadow: none ; }
```

```

1478
1479 div.floatrow { text-align: center; }
1480
1481 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1482
1483 div.floatfoot { font-size: .85em ;
1484     border-top: 1px solid silver ; line-height: 1.2 ; }
1485
1486 figcaption , .lstlistingtitle {
1487     font-size: .85em ;
1488     text-align: center ;
1489     font-weight: bold ;
1490     margin-top: 1ex ;
1491     margin-bottom: 1ex ;
1492 }
1493
1494 figure.subfigure figcaption, figure.subtable figcaption {
1495     border-bottom: none ; background: none ;
1496 }
1497
1498 div.nonfloatcaption {
1499     margin: 1ex auto 1ex auto ;
1500     font-size: .85em ;
1501     text-align: center ;
1502     font-weight: bold ;
1503 }
1504
1505 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1506 figure div.floatrow div.minipage figcaption {
1507     border: none ;
1508     background: none ;
1509 }
1510
1511
1512 table {
1513     margin: 1ex auto 1ex auto ;
1514     border-collapse: separate ;
1515     border-spacing: 0px ;
1516     line-height: 1.3 ;
1517 }
1518
1519 tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
1520     margin-bottom: 0ex ; } /* for \hline */
1521
1522 tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
1523     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1524
1525 td {padding: 1ex .5em 1ex .5em ;}
1526
1527 table td.tdl { text-align: left ; vertical-align: middle ; }

```

```

1528 table td.tdc { text-align: center ; vertical-align: middle ; }
1529 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1530 table td.tdbang { text-align: center ; vertical-align: middle ; }
1531 table td.tdr { text-align: right ; vertical-align: middle ; }
1532 table td.tdp { text-align: left ; vertical-align: bottom ; }
1533 table td.tdm { text-align: left ; vertical-align: middle ; }
1534 table td.tdb { text-align: left ; vertical-align: top ; }
1535 table td.tdP { text-align: center ; vertical-align: bottom ; }
1536 table td.tdM { text-align: center ; vertical-align: middle ; }
1537 table td.tdB { text-align: center ; vertical-align: top ; }
1538
1539 table td.tvertbarl { border-left: 1px solid black }
1540 table td.tvertbarr { border-right: 1px solid black }
1541
1542
1543 /* for cmidrules: */
1544 table td.tdrule {
1545     border-top: 1px solid #A0A0A0 ;
1546 }
1547
1548 table td.tdrulel {
1549     border-top-left-radius:.5em ;
1550     border-top: 1px solid #A0A0A0 ;
1551 }
1552
1553 table td.tdruler {
1554     border-top-right-radius:.5em ;
1555     border-top: 1px solid #A0A0A0 ;
1556 }
1557
1558 table td.tdrulelr {
1559     border-top-left-radius:.5em ;
1560     border-top-right-radius:.5em ;
1561     border-top: 1px solid #A0A0A0 ;
1562 }
1563
1564
1565 /* Margins of paragraphs inside table cells: */
1566 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1567     padding-bottom: 1ex ; margin: 0ex ; }
1568 td.tdm p , td.tdmrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1569     padding-bottom: 1ex ; margin: 0ex ; }
1570 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1571     padding-bottom: 1ex ; margin: 0ex ; }
1572
1573 td.tdp , td.tdprule , td.tdP , td.tdPrule
1574     { padding: 0ex .5em 0ex .5em ; }
1575 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1576     { padding: 0ex .5em 0ex .5em ; }
1577 td.tdb , td.tdbrule , td.tdB , td.tdBrule

```

```
1578     { padding: 0ex .5em 0ex .5em ; }
1579
1580
1581 /* table notes: */
1582 .tnotes {
1583     margin: 0ex 5% 1ex 5% ;
1584     padding: 0.5ex 1em 0.5ex 1em;
1585     font-size:.85em;
1586     text-align: left ;
1587 }
1588
1589 .tnotes dl dt p {margin-bottom:0px;}
1590
1591 .tnoteitemheader {margin-right: 1em;}
1592
1593
1594 /* for bigdelim */
1595 .ldelim, .rdelim { font-size: 200% }
1596
1597
1598 /* center, flushleft, flushright environments */
1599 div.center{text-align:center;}
1600 div.center table {margin-left:auto;margin-right:auto;}
1601 div.flushleft{text-align:left;}
1602 div.flushleft table {margin-left:0em ; margin-right:auto;}
1603 div.flushright{text-align:right;}
1604 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1605
1606
1607 /* Fancybox */
1608 div.Btrivlist table tr td { padding: .2ex 0em ; }
1609
1610
1611 /* program listing callouts: */
1612 span.callout {
1613     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1614         Geneva, Verdana, sans-serif ;
1615     border-radius: .5em;
1616     background-color:black;
1617     color:white;
1618     padding:0px .25em 0px .25em;
1619     margin: 0 ;
1620     font-weight: bold;
1621     font-size:.72em ;
1622 }
1623
1624 div.programlisting pre.verbatim span.callout{
1625     font-size: .85em ;
1626 }
1627
```

```
1628
1629
1630
1631
1632 div.published
1633 {
1634     text-align: center ;
1635     font-variant: normal ;
1636     font-style: italic ;
1637     font-size: 1em ;
1638     margin: 3ex 0em 3ex 0em ;
1639 }
1640
1641 div.subtitle
1642 {
1643     text-align: center ;
1644     font-variant: normal ;
1645     font-style: italic ;
1646     font-size: 1.25em ;
1647     margin: 3ex 0em 3ex 0em ;
1648 }
1649
1650 div.subtitle p { margin: 1ex ; }
1651
1652 div.author
1653 {
1654     font-variant: normal ;
1655     font-style: normal ;
1656     font-size: 1em ;
1657     margin: 3ex 0em 3ex 0em ;
1658 }
1659
1660 div.oneauthor {
1661     display: inline-block ;
1662     margin: 3ex 1em 0ex 1em ;
1663 }
1664
1665 /*
1666 div.author table {
1667     margin: 3ex auto 0ex auto ;
1668     background: none ;
1669 }
1670
1671 div.author table tbody tr td { padding: .25ex ; }
1672 */
1673
1674 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1675
1676 div.titledate {
1677     text-align: center ;
```

```
1678     font-size: .85em ;
1679     font-style: italic;
1680     margin: 6ex 0em 6ex 0em ;
1681 }
1682
1683
1684 nav.topnavigation{
1685     text-align: left ;
1686     padding: 0.5ex 1em 0.5ex 1em ;
1687 /*     margin: 2ex 0em 3ex 0em ; */
1688     margin: 0 ;
1689     border-bottom: 1px solid silver ;
1690     border-top: 1px solid silver ;
1691     clear:right ;
1692 }
1693
1694 nav.botnavigation{
1695     text-align: left ;
1696     padding: 0.5ex 1em 0.5ex 1em ;
1697 /*     margin: 3ex 0em 2ex 0em ; */
1698     margin: 0 ;
1699     border-top: 1px solid silver ;
1700     border-bottom: 1px solid silver ;
1701     clear:right ;
1702 }
1703
1704
1705 header{
1706     line-height: 1.2 ;
1707     font-size: 1em ;
1708 /*     border-bottom: 2px solid silver ; */
1709     margin: 0px ;
1710     padding: 0ex 1em 0ex 1em ;
1711     text-align:center ;
1712 }
1713
1714 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1715
1716
1717 footer{
1718     font-size: .85em ;
1719     line-height: 1.2 ;
1720     margin-top: 1ex ;
1721     border-top: 2px solid silver ;
1722     padding: 2ex 1em 2ex 1em ;
1723     clear:right ;
1724     text-align:left ;
1725 }
1726
1727
```



```
1728 a.linkhome { font-weight:bold ; font-size: 1em ;}
1729
1730
1731 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1732
1733 img.lateximage{
1734     padding: 0px 0px 0px 0px ;
1735     box-shadow: none ;
1736     border: none ;
1737     background: none ;
1738     margin: 0px 0px -.15ex 0px ;
1739     /* pdfcrop leaves a slight margin, adjust to baseline */
1740     max-width: 100% ;
1741     border-radius: 0ex ;
1742     border: none ;
1743 }
1744
1745
1746
1747 nav.sidetoc {
1748     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1749         "Lucida Bright", Georgia, serif;
1750     float:right ;
1751     width: 20%;
1752     border-left: 1px solid silver;
1753     border-top: 1px solid silver;
1754     border-bottom: 1px solid silver;
1755     /* border-top: 2px solid #808080 ; */
1756     background: #FAF7F4 ;
1757     padding: 2ex 0em 2ex 1em ;
1758     margin: 0ex 0em 2ex 1em ;
1759     font-size:.9em ;
1760     border-radius: 20px 0px 0px 20px ;
1761 }
1762
1763 div.sidetoccontents {
1764     /* border-top: 1px solid silver ; */
1765     overflow-y: auto ;
1766     width: 100% ;
1767     text-align: left ;
1768 }
1769
1770
1771 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1772     text-indent: 0 ; }
1773
1774 nav.sidetoc p a {color:black ; font-size: .7em ;}
1775
1776 div.sidetocctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1777     border-bottom: 1px solid silver ; }
```

```
1778
1779 nav.sidetoc a:hover {text-decoration: underline ; }
1780
1781
1782
1783 section.textbody { margin: 0ex 1em 0ex 1em ;}
1784
1785
1786 div.multicolsheading { -webkit-column-span: all;
1787     -moz-column-span: all; column-span: all; }
1788 div.multicols { -webkit-columns: 3 380px ;
1789     -moz-columns: 3 380px ; columns: 3 380px ; }
1790 div.multicols p {margin-top: 0ex}
1791
1792
1793
1794 /* Used to support algorithmicx: */
1795 span.floatright { float: right ; }
1796
1797
1798
1799
1800 /* Native LaTeX theorems: */
1801
1802 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1803 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1804
1805
1806 /* theorem, amsthm, and ntheorem packages */
1807
1808 span.theoremheader,
1809 span.theoremheaderplain,
1810 span.theoremheaderdefinition,
1811 span.theoremheaderbreak,
1812 span.theoremheadermarginbreak,
1813 span.theoremheaderchangebreak,
1814 span.theoremheaderchange,
1815 span.theoremheadermargin
1816 {
1817     font-style:normal ; font-weight: bold ; margin-right: 1em ;
1818 }
1819
1820 span.amsthmnameplain,
1821 span.amsthmnamedefinition,
1822 span.amsthmnumberplain,
1823 span.amsthmnumberdefinition
1824 {
1825     font-style:normal ; font-weight: bold ;
1826 }
1827
```

```
1828
1829 span.amsthmnameremark,
1830 span.amsthmnumberremark
1831 {font-style:italic ; font-weight: normal ; }
1832
1833
1834 span.amsthmnoteplain,
1835 span.amsthmnotedefinition
1836 {font-style:normal ;}
1837
1838
1839 span.theoremheaderremark,
1840 span.theoremheaderproof,
1841 span.amsthmproofname
1842 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1843
1844 span.theoremheadersc
1845 {
1846     font-style:normal ;
1847     font-variant: small-caps ;
1848     font-weight: normal ;
1849     margin-right: 1em ;
1850 }
1851
1852 .theoremdemark {float:right}
1853
1854 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1855 div.theorembodybreak, div.theorembodynonumberbreak,
1856 div.theorembodymarginbreak,
1857 div.theorembodychangebreak,
1858 div.theorembodychange,
1859 div.theorembodymargin
1860 {
1861     font-style:italic;
1862     margin-top: 3ex ; margin-bottom: 3ex ;
1863 }
1864
1865 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1866 div.theorembodyplainupright, nonumberplainuprightsc,
1867 div.amsthmbodydefinition, div.amsthmbodyremark,
1868 div.amsthmproof
1869 {
1870     font-style: normal ;
1871     margin-top: 3ex ; margin-bottom: 3ex ;
1872 }
1873
1874 span.amsthmnoteremark {}
1875
1876
1877
```

```
1878 /*
1879 For CSS LaTeX and related logos:
1880 Based on:
1881 http://edward.oconnor.cx/2007/08/tex-poshlet
1882 http://nitens.org/taraborelli/texlogo
1883 */
1884
1885 .latexlogofont {
1886     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1887         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1888     font-variant: normal ;
1889 }
1890
1891 .latexlogo {
1892     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1893         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1894     letter-spacing: .03em ;
1895     font-size: 1.1em;
1896 }
1897
1898 .latexlogo sup {
1899     text-transform: uppercase;
1900     letter-spacing: .03em ;
1901     font-size: 0.85em;
1902     vertical-align: 0.15em;
1903     margin-left: -0.36em;
1904     margin-right: -0.15em;
1905 }
1906
1907 .latexlogo sub {
1908     text-transform: uppercase;
1909     vertical-align: -0.5ex;
1910     margin-left: -0.1667em;
1911     margin-right: -0.125em;
1912     font-size: 1em;
1913 }
1914
1915 .xetexlogo {
1916     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
1917         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1918     letter-spacing: .03em ;
1919     font-size: 1.1em;
1920 }
1921
1922 /* A smaller gap between Xe and Tex v.s. LaTeX: */
1923 .xetexlogo sub {
1924     text-transform: uppercase;
1925     vertical-align: -0.5ex;
1926     margin-left: -0.0667em;
1927     margin-right: -0.2em;
```

```
1928 font-size: 1em;
1929 letter-spacing: .03em ;
1930 }
1931
1932 /* A large gap between Xe and LaTeX v.s. TeX: */
1933 .xelatexlogo sub {
1934     text-transform: uppercase;
1935     vertical-align: -0.5ex;
1936     margin-left: -0.0667em;
1937     margin-right: -.05em;
1938     font-size: 1em;
1939     letter-spacing: .03em ;
1940 }
1941
1942 .amslogo {
1943     font-family: "TeXGyreChorus","URW Chancery L",
1944         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
1945         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
1946         "Hoefler Text", Times, "Times New Roman", serif;
1947     font-style: italic;
1948 }
1949
1950 .lyxlogo {
1951     font-family: "URW Classico", Optima, "Linux Biolinum O",
1952         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
1953         Verdana, sans-serif ;
1954 }
1955
1956
1957
1958
1959 /* Only display top and bottom navigation if a small screen: */
1960 /* Hide the sidetoc if a small screen: */
1961 nav.topnavigation { display:none; }
1962 nav.botnavigation { display:none; }
1963
1964 @media screen and (max-width: 45em) {
1965     /*     nav.sidetoc {display:none;} */
1966     nav.sidetoc {
1967         float: none ;
1968         width: 100% ;
1969         margin: 5ex 0px 5ex 0px ;
1970         padding: 0 ;
1971         border-radius: 0 ;
1972         border-bottom: 1px solid black ;
1973         border-top: 1px solid black ;
1974         box-shadow: none ;
1975     }
1976     /*     nav.topnavigation { display:block } */
1977     nav.botnavigation { display:block }
```

```

1978     .marginpar {
1979         max-width: 100%;
1980         float: none;
1981         display: block ;
1982         margin: 1ex 1em 1ex 1em ;
1983     }
1984 }
1985
1986 @media print {
1987     body {
1988         font-family: "Linux Libertine O",
1989         "DejaVu Serif", "Bitstream Vera Serif",
1990         "Liberation Serif", "Nimbus Roman No 9 L",
1991         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1992     }
1993     nav.sidetoc { display: none; }
1994     nav.topnavigation { display: none; }
1995     nav.botnavigation { display: none; }
1996 }
1997
1998 @media handheld {
1999     nav.sidetoc { display: none; }
2000     nav.topnavigation { display: block }
2001     nav.botnavigation { display: block }
2002 }
2003
2004 @media projection {
2005     nav.sidetoc { display: none; }
2006     nav.topnavigation { display: block }
2007     nav.botnavigation { display: block }
2008 }
2009 \end{filecontents*}
2010 % \end{Verbatim}% for syntax highlighting
2011 \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.

```

2012 \begin{warpprint}
2013 \begin{filecontents*}{lwarp_sagebrush.css}
2014 @import url("lwarp.css") ;
2015
2016

```

```
2017 A:link {color:#105030 ; text-decoration: none ; }
2018 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2019 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2020 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2021
2022
2023
2024 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2025 {
2026     font-family: "URW Classico", Optima, "Linux Biolinum O",
2027         "Linux Libertine O", "Liberation Serif",
2028         "Nimbus Roman No 9 L", "FreeSerif",
2029         "Hoefler Text", Times, "Times New Roman", serif;
2030     font-variant: small-caps ;
2031     font-weight: normal ;
2032     color: #304070 ;
2033     text-shadow: 2px 2px 3px #808080;
2034 }
2035
2036 h1 { /* title of the entire website, used on each page */
2037     font-variant: small-caps ;
2038     color: #304070 ;
2039     text-shadow: 2px 2px 3px #808080;
2040     background-color: #F7F7F0 ;
2041     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2042 }
2043
2044 h1 {
2045     border-bottom: 1px solid #304070;
2046     border-top: 2px solid #304070;
2047 }
2048
2049 h2 {
2050     border-bottom: 1px solid #304070;
2051     border-top: 2px solid #304070;
2052     background-color: #F7F7F0 ;
2053     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2054 }
2055
2056
2057
2058 div.abstract {
2059     background: #f5f5eb ;
2060     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2061
2062     border: 1px solid silver;
2063     border-radius: 1em ;
2064 }
2065
2066 div.abstract dl {line-height:1.5;}
```

```
2067 div.abstract dt {color:#304070;}
2068
2069 div.abstracttitle{
2070     font-family: "URW Classico", Optima, "Linux Biolinum O",
2071         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2072         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2073     font-weight:bold;
2074     font-variant: small-caps ;
2075     font-size:1.5em;
2076     border-bottom: 1px solid silver ;
2077     color: #304070 ;
2078     text-align: center ;
2079     text-shadow: 1px 1px 2px #808080;
2080 }
2081
2082 span.abstracrunintitle{
2083     font-family: "URW Classico", Optima, "Linux Biolinum O",
2084         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2085         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2086     font-weight:bold;
2087 }
2088
2089
2090 div.epigraph {
2091     background: #f5f5eb ;
2092     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2093
2094     border: 1px solid silver ;
2095     border-radius: 1ex ;
2096     box-shadow: 3px 3px 3px #808080 ;
2097 }
2098
2099
2100 .example {
2101     background-color: #f5f5eb ;
2102     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2103
2104 }
2105
2106 div.exampletitle{
2107     font-family: "URW Classico", Optima, "Linux Biolinum O",
2108         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2109         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2110     font-weight:bold;
2111     font-variant: small-caps ;
2112     border-bottom: 1px solid silver ;
2113     color: #304070 ;
2114     text-align: center ;
2115     text-shadow: 1px 1px 2px #808080;
2116 }
```



```
2117
2118
2119 .sidebar {
2120     background-color: #f5f5eb ;
2121     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2122
2123 }
2124
2125 div.sidebartitle{
2126     font-family: "URW Classico", Optima, "Linux Biolinum O",
2127         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2128         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2129     font-weight:bold;
2130     font-variant: small-caps ;
2131     border-bottom: 1px solid silver ;
2132     color: #304070 ;
2133     text-align: center ;
2134     text-shadow: 1px 1px 2px #808080;
2135 }
2136
2137
2138 .fancyvrblabel {
2139     font-family: "URW Classico", Optima, "Linux Biolinum O",
2140         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2141         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2142     font-weight:bold;
2143     font-variant: small-caps ;
2144     font-size: 1.5em ;
2145     color: #304070 ;
2146     text-align: center ;
2147     text-shadow: 1px 1px 2px #808080;
2148 }
2149
2150
2151
2152 div.minipage {
2153     background-color: #eeeeee7 ;
2154     border: 1px solid silver ;
2155     border-radius: 1ex ;
2156 }
2157
2158 section.textbody > div.minipage {
2159     box-shadow: 3px 3px 3px #808080 ;
2160 }
2161
2162 div.fboxBlock div.minipage { box-shadow: none ; }
2163
2164 .framed .minipage , .framedleftbar .minipage {
2165     border: none ;
2166     background: none ;
```

```
2167     padding: 0ex ;
2168     margin: 0ex ;
2169 }
2170
2171 figure.figure .minipage, figcaption .minipage { border: none; }
2172
2173 div.marginblock div.minipage { border: none; }
2174
2175 figure , div.marginblock {
2176     background-color: #eeeeee7 ;
2177     border: 1px solid silver ;
2178     border-radius: 1ex ;
2179     box-shadow: 3px 3px 3px #808080 ;
2180 }
2181
2182 figure figure {
2183     border: 1px solid silver ;
2184     margin: 0em ;
2185     box-shadow: none ;
2186 }
2187
2188 /*
2189 figcaption {
2190     border-top: 1px solid silver ;
2191     border-bottom: 1px solid silver ;
2192     background-color: #e8e8e8 ;
2193 }
2194 */
2195
2196
2197 div.table {
2198     box-shadow: 3px 3px 3px #808080 ;
2199 }
2200
2201 /*
2202 .tnotes {
2203     background: #e8e8e8;
2204     border: 1px solid silver;
2205 }
2206 */
2207
2208
2209 nav.topnavigation{
2210     background-color: #b0b8b0 ;
2211     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2212 }
2213
2214 nav.botnavigation{
2215     background-color: #b0b8b0 ;
2216     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
```

```

2217 }
2218
2219
2220
2221 header{
2222     background-color: #F7F7F0 ;
2223     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2224 }
2225
2226 footer{
2227     background-color: #F7F7F0 ;
2228     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2229 }
2230
2231
2232
2233 nav.sidetoc {
2234     background-color: #F7F7F0 ;
2235     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2236     box-shadow: 3px 3px 3px #808080 ;
2237     border-radius: 0px 0px 0px 20px ;
2238 }
2239
2240 div.sidetoc title {color: #304070 ; }
2241
2242 nav.sidetoc a: hover {
2243     color: #006000 ;
2244     text-decoration: none ;
2245     text-shadow: 0px 0px 2px #a0a0a0;
2246 }
2247
2248
2249 @media screen and (max-width: 45em) {
2250     nav.sidetoc { border-radius: 0 ; }
2251 }
2252
2253
2254 \end{filecontents*}
2255 % \end{Verbatim}% for syntax highlighting
2256 \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.

```

2257 \begin{warpprint}
2258 \begin{filecontents*}{lwarp_formal.css}
2259 @import url("lwarp.css") ;
2260
2261
2262
2263 A:link {color:#802020 ; text-decoration:none; }
2264 A:visited {color:#802020 ; text-shadow:none ;}
2265 A:hover {color:#400000 ; text-shadow:none ;}
2266 A:active {color:#C00000 ; text-shadow:none ;}
2267
2268
2269 body {
2270     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2271         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2272         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2273         "Times New Roman", serif;
2274     background: #fffcf5;
2275 }
2276
2277 span.textrm {
2278     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2279         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2280         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2281         "Times New Roman", serif;
2282 }
2283
2284 span.textsf {
2285     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2286         Geneva, Verdana, sans-serif ;
2287 }
2288
2289
2290
2291 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2292 {
2293     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2294         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2295         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2296         "Times New Roman", serif;
2297     color: #800000 ;
2298     text-shadow: none ;
2299 }
2300
2301 h1, h2 {
2302     background-color: #fffcf5 ;

```

```
2303     background-image: none ;
2304     border-bottom: 1px solid #808080;
2305     border-top: 2px solid #808080;
2306 }
2307
2308 div.abstracttitle {
2309     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2310         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2311         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2312         "Times New Roman", serif;
2313     color: black ;
2314     text-shadow: none ;
2315 }
2316
2317 span.abstracrunintitle {
2318     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2319         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2320         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2321         "Times New Roman", serif;
2322     color: black ;
2323     text-shadow: none ;
2324 }
2325
2326 div.abstract { font-size: 100% }
2327
2328 .sidebar {
2329     background: #fffcf5;
2330     background-image: none ;
2331     margin: 2em 5% 2em 5%;
2332     padding: 0.5em 1em;
2333     border: none ;
2334     border-top : 1px solid silver;
2335     border-bottom : 1px solid silver;
2336     font-size: 90% ;
2337 }
2338
2339 div.sidebartitle{
2340     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2341         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2342         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2343         "Times New Roman", serif;
2344     color: #800000 ;
2345     text-shadow: none ;
2346     border: none ;
2347 }
2348
2349 .example {
2350     background: #fffcf5;
2351     background-image: none ;
2352     margin: 2em 5% 2em 5%;
```

```
2353 padding: 0.5em 1em;
2354 border: none ;
2355 border-top : 1px solid silver;
2356 border-bottom : 1px solid silver;
2357 }
2358
2359 div.exampletitle{
2360     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2361         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2362         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2363         "Times New Roman", serif;
2364     color: #800000 ;
2365     text-shadow: none ;
2366     border: none ;
2367 }
2368
2369 div.fancyvrlabel{
2370     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2371         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2372         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2373         "Times New Roman", serif;
2374     color: #800000 ;
2375     text-shadow: none ;
2376     border: none ;
2377 }
2378
2379
2380
2381 .verse {
2382     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2383         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2384         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2385         "Times New Roman", serif;
2386 }
2387
2388
2389 figure {
2390     margin: 3ex 5% 3ex 5% ;
2391     padding: 1ex 1em 1ex 1em ;
2392     background-color: #fffcf5 ;
2393     overflow-x: auto ;
2394     border: none ;
2395     /* border-top: 1px solid silver; */
2396     /* border-bottom: 1px solid silver; */
2397 }
2398
2399
2400 figcaption , .lstlisting {
2401     border: none ;
2402     /* border-top: 1px solid silver ; */
```

```
2403 /*      border-bottom: 1px solid silver ; */
2404      background-color: #fffcf5 ;
2405 }
2406
2407 .tnotes {
2408     background: #fffcf5 ;
2409 }
2410
2411 .theorem {
2412     background: none ;
2413 }
2414
2415 .minipage {
2416     background-color: #fffcf5 ;
2417     border: none ;
2418 }
2419
2420 div.floatrow figure { border: none ; }
2421
2422 figure figure { border: none ; }
2423
2424
2425 nav.toc, nav.lof, nav.lot, nav.lol {
2426     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2427         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2428         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2429         "Times New Roman", serif;
2430 }
2431
2432 nav.sidetoc {
2433     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2434         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2435         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2436         "Times New Roman", serif;
2437     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
2438     border-radius: 0px 0px 0px 20px ;
2439 }
2440
2441 div.sidetoc title{
2442     color: #800000 ;
2443 }
2444
2445 header{
2446     background-color: #e0e0e0 ;
2447     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2448     text-align:center ;
2449 }
2450
2451 footer{
2452     background-color: #e0e0e0 ;
```

```

2453     background-image: linear-gradient(to bottom, #ffcf5, #b0b0b0);
2454     padding: 2ex 1em 2ex 1em ;
2455     clear:right ;
2456     text-align:left ;
2457 }
2458
2459 nav.botnavigation {
2460     background: #dedcd5 ;
2461     border-top: 1px solid black ;
2462 }
2463 \end{filecontents*}
2464 % \end{Verbatim}% for syntax highlighting
2465 \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.

```

2466 \begin{warpprint}
2467 \begin{filecontents*}{sample_project.css}
2468 /* ( --- Start of project.css --- ) */
2469 /* A sample project-specific CSS file for lwarp --- ) */
2470
2471 /* Load default lwarp settings: */
2472 @import url("lwarp.css") ;
2473 /* or lwarp_formal.css, lwarp_sagebrush.css */
2474
2475 /* Project-specific CSS setting follow here. */
2476 /* . . . */
2477
2478 /* ( --- End of project.css --- ) */
2479 \end{filecontents*}
2480 % \end{Verbatim}% for syntax highlighting
2481 \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.


```

2482 \begin{warpprint}
2483 \begin{filecontents*}{lwarp.xdy}
2484 (require "tex/inputenc/latin.xdy")
2485 (merge-rule "\\PS *" "Postscript")
2486 (require "texindy.xdy")
2487 (require "page-ranges.xdy")
2488 (require "book-order.xdy")
2489 (markup-locref :open "\hyperindexref{" :close "{")
2490 \end{filecontents*}
2491 % \end{Verbatim}% for syntax highlighting
2492 \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.

```

2493 \begin{warpprint}
2494 \begin{filecontents*}{lwarp_mathjax.txt}
2495 <!-- https://groups.google.com/forum/#!topic/
2496                               mathjax-users/jUtewUcE2bY -->
2497 <script type="text/x-mathjax-config">
2498 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2499     var seteqsectionDefault = {name: "", num: 0};
2500     var seteqsections = {}, seteqsection = seteqsectionDefault;
2501     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2502     var AMS = MathJax.Extension["TeX/AMSmath"];
2503     TEX.Definitions.Add({
2504       macros: {
2505         seteqsection: "mySection",
2506         seteqnumber: "mySetEqNumber"
2507       }
2508     });
2509
2510     PARSE.Augment({
2511       mySection: function (name) {
2512         seteqsection.num = AMS.number;
2513         var n = this.GetArgument(name);
2514         if (n === "") {
2515           seteqsection = seteqsectionDefault;
2516         } else {
2517           if (!seteqsections["_"+n])
2518             seteqsections["_"+n] = {name:n, num:0};
2519           seteqsection = seteqsections["_"+n];
2520         }

```

```

2521     AMS.number = seteqsection.num;
2522   },
2523   mySetEqNumber: function (name) {
2524     var n = this.GetArgument(name);
2525     if (!n || !n.match(/^ *[0-9]+ *$/))
2526       n = ""; else n = parseInt(n)-1;
2527     <!-- $ syntax highlighting -->
2528     if (n === "" || n < 1)
2529       TEX.Error
2530       ("Argument to "+name+" should be a positive integer");
2531     AMS.number = n;
2532   }
2533   });
2534   MathJax.Hub.Config({
2535     TeX: {
2536       equationNumbers: {
2537         formatTag: function (n)
2538           {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
2539         formatID: function (n) {
2540           n = (seteqsection.name+'.'+n).replace
2541             (/[:'>&]/g,"").replace(/\./,"");
2542           return 'mjx-eqn-' + n;
2543         }
2544       }
2545     }
2546   });
2547 });
2548 </script>
2549
2550 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2551 <script type="text/x-mathjax-config">
2552   MathJax.Ajax.config.path["Contrib"] =
2553     "https://cdn.mathjax.org/mathjax/contrib";
2554 </script>
2555
2556 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
2557       tree/master/siunitx -->
2558 <script type="text/x-mathjax-config">
2559   MathJax.Hub.Config({
2560     extensions: ["tex2jax.js", "[Contrib]/siunitx/siunitx.js"],
2561     jax: ["input/TeX", "output/HTML-CSS"],
2562     tex2jax: {inlineMath: [["$", "$"], ["\\(", "\\)"]]},
2563     TeX: {extensions: ["AMSMath.js", "AMSsymbols.js", "sinuitx.js"]}
2564   });
2565 </script>
2566
2567 <script type="text/x-mathjax-config">
2568   MathJax.Hub.Config({
2569     TeX: {
2570       equationNumbers: {

```

```

2571         autoNumber: "AMS"
2572     }
2573 }
2574 });
2575 </script>
2576
2577 <!-- Alternative CDN provider: -->
2578 <script type="text/javascript" async
2579 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2580 </script>
2581
2582 <!-- No longer supported after April 30, 2017: -->
2583 <!--
2584 <script
2585   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2586 </script>
2587 -->
2588
2589 \end{filecontents*}
2590 % \end{Verbatim}% for syntax highlighting
2591 \end{warpprint}

```

30.10 lwarpmk option

The following is only generated if the lwarpmk option was given to lwarp.

```

2592 \begin{LWR@createlwarpmk}

```

Prog lwarpmk Creates a local copy of lwarpmk:

```

2593 \begin{filecontents*}{lwarpmk.lua}
2594 #!/usr/bin/env texlua
2595
2596 -- Copyright 2016-2017 Brian Dunn
2597
2598 -- Print the usage of the lwarpmk command:
2599
2600 printversion = "v0.41"
2601
2602 function printhelp ()
2603 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2604 end
2605
2606 function printusage ()
2607 print ( [[
2608
2609 lwarpmk print [project]: Compile the print version if necessary.

```

```

2610 lwarpmk print1 [project]: Forced single compile of the print version.
2611 lwarpmk printindex [project]: Process the index for the print version.
2612 lwarpmk printglossary [project]: Process the glossary for the print version.
2613 lwarpmk html [project]: Compile the HTML version if necessary.
2614 lwarpmk html1 [project]: Forced single compile of the HTML version.
2615 lwarpmk htmlindex [project]: Process the index for the html version.
2616 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2617 lwarpmk again [project]: Touch the source code to trigger recompiles.
2618 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2619 lwarpmk pdftohtml [project]:
2620     For use with latexmk or a Makefile:
2621     Convert project_html.pdf to project_html.html and
2622     individual HTML files.
2623 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
2624 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2625 lwarpmk -h: Print this help message.
2626 lwarpmk --help: Print this help message.
2627
2628 ]] )
2629 printconf ( )
2630 end
2631
2632 -- Print the format of the configuration file lwarpmk.conf:
2633
2634 function printconf ( )
2635 print ( [[
2636 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2637 --
2638 opsystem = "Unix"      (or "Windows")
2639 latexname = "pdflatex" (or "lualatex", or "xelatex")
2640 sourcename = "projectname" (the source-code filename w/o .tex)
2641 homehtmlfilename = "index" (or perhaps the project name)
2642 htmlfilename = ""      (or "projectname" - filename prefix)
2643 latexmk = "false"      (or "true" to use latexmk to build PDFs)
2644 language = "english"   (use a language supported by xindy)
2645 xdyfile = "lwarp.xdy"  (or a custom file based on lwarp.xdy)
2646 --
2647 Filenames must contain only letters, numbers, underscore, or dash.
2648 Values must be in "quotes".
2649
2650 ]] ) ;
2651 end
2652
2653
2654 -- Split one large sourcefile into a number of files,
2655 -- starting with destfile.
2656 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2657
2658 function splitfile (destfile,sourcefile)
2659 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;

```

```

2660 local sfile = io.open(sourcefile)
2661 io.output(destfile)
2662 for line in sfile:lines() do
2663   i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
2664   if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2665     io.output(newfilename) ;
2666   else -- not a splitpoint
2667     io.write (line .. "\n") ;
2668   end
2669 end -- do
2670 io.close(sfile)
2671 end -- function
2672
2673 -- Incorrect value, so print an error and exit.
2674
2675 function cvalueerror ( line, linenum , cvalue )
2676   print ( linenum .. " : " .. line ) ;
2677   print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2678   printconf () ;
2679   os.exit(1) ;
2680 end
2681
2682 -- Load settings from the project's "lwarpmk.conf" file:
2683
2684 function loadconf ()
2685 -- Default configuration filename:
2686 local conffile = "lwarpmk.conf"
2687 -- Optional configuration filename:
2688 if arg[2] ~= nil then conffile = arg[2].."lwarpmk.conf" end
2689 -- Default language:
2690 language = "english"
2691 -- Default xdyfile:
2692 xdyfile = "lwarp.xdy"
2693 -- Verify the file exists:
2694 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2695   print("lwarpmk: " .. conffile .. " does not exist.")
2696   print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2697   printhelp () ;
2698   os.exit(1) -- exit the entire lwarpmk script
2699 else -- file exists
2700 -- Read the file:
2701   print ("lwarpmk: Reading " .. conffile .. ".")
2702   local cfile = io.open(conffile)
2703 -- Scan each line:
2704   local linenum = 0
2705   for line in cfile:lines() do -- scan lines
2706     linenum = linenum + 1
2707     i,j,cvarname,cvalue = string.find (line,"([%w-_*])%s*=%s*\"([%w%-_.]*)\"") ;
2708 -- Error if incorrect enclosing characters:
2709     if ( i == nil ) then

```

```
2710 print ( linenum .. " : " .. line ) ;
2711 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2712 printconf ( ) ;
2713 os.exit(1) ;
2714 end
2715 if ( cvarname == "opssystem" ) then
2716     -- Verify choice of opssystem:
2717     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2718         opssystem = cvalue
2719     else
2720         cvalueerror ( line, linenum , cvalue )
2721     end
2722 elseif ( cvarname == "latexname" ) then
2723     -- Verify choice of LaTeX compiler:
2724     if (
2725         (cvalue == "pdflatex") or
2726         (cvalue == "xelatex") or
2727         (cvalue == "lualatex")
2728     ) then
2729         latexname = cvalue
2730     else
2731         cvalueerror ( line, linenum , cvalue )
2732     end
2733 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2734 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2735 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2736 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2737 elseif ( cvarname == "language" ) then language = cvalue
2738 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2739 else
2740 print ( linenum .. " : " .. line ) ;
2741 print ( "lwarpmk: Incorrect variable name \" " .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2742 printconf ( ) ;
2743 os.exit(1) ;
2744 end
2745 end -- do scan lines
2746 io.close(cfile)
2747 end -- file exists
2748 -- Select some operating-system commands:
2749 if opssystem=="Unix" then -- For Unix / Linux / Mac OS:
2750 rmname = "rm"
2751 mvname = "mv"
2752 touchnamepre = "touch"
2753 touchnamepost = ""
2754 dirslash = "/"
2755 opquote= "\""
2756 elseif opssystem=="Windows" then -- For Windows
2757 rmname = "DEL"
2758 mvname = "MOVE"
2759 touchnamepre = "COPY /b"
```

```
2760 touchnamepost = "+,,"
2761 dirslash = "\\\"
2762 opquote= "\"\"
2763 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2764 end --- for Windows
2765
2766 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2767 if ( latexname == "pdflatex" ) then
2768 xindycmd = "texindy -C utf8"
2769 glossarycmd = "xindy -C utf8"
2770 else
2771 xindycmd = "xindy -M texindy -C utf8"
2772 glossarycmd = "xindy -C utf8"
2773 end
2774
2775 end -- loadconf
2776
2777
2778 function refreshdate ()
2779 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2780 end
2781
2782
2783 -- Scan the LaTeX log file for the phrase "Rerun to get",
2784 -- indicating that the file should be compiled again.
2785 -- Return true if found.
2786
2787 function reruntoget (filesorce)
2788 local fsource = io.open(filesorce)
2789 for line in fsource:lines() do
2790 if ( string.find(line,"Rerun to get") ~= nil ) then
2791     io.close(fsource)
2792     return true
2793 end
2794 end
2795 io.close(fsource)
2796 return false
2797 end
2798
2799
2800 -- Compile one time, return true if should compile again.
2801 -- fsuffix is "" for print, "_html" for HTML output.
2802
2803 function onetime (fsuffix)
2804 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2805 err = os.execute(
2806 --     "echo " ..
2807     latexname .. " " .. sourcename..fsuffix )
2808 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2809 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
```

```
2810 end
2811
2812
2813 -- Compile up to five times.
2814 -- fsuffix is "" for print, "_html" for HTML output
2815
2816 function manytimes (fsuffix)
2817 if onetime(fsuffix) == true then
2818 if onetime(fsuffix) == true then
2819 if onetime(fsuffix) == true then
2820 if onetime(fsuffix) == true then
2821 if onetime(fsuffix) == true then
2822 end end end end end
2823 end
2824
2825 -- Exit if the given file does not exist.
2826
2827 function verifyfileexists (filename)
2828 if (lfs.attributes ( filename , "modification" ) == nil ) then
2829 print ( "lwarpmk: " .. filename .. " not found." );
2830 os.exit (1) ;
2831 end
2832 end
2833
2834
2835 -- Convert <project>_html.pdf into HTML files:
2836
2837 function pdftohtml ()
2838     -- Convert to text:
2839     print ("lwarpmk: Converting " .. sourcename
2840         .. "_html.pdf to " .. sourcename .. "_html.html")
2841     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2842         .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2843     -- Split the result into individual HTML files:
2844     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2845 end
2846
2847
2848 -- Remove auxiliary files:
2849
2850 function removeaux ()
2851     os.execute ( rmname .. " " ..
2852         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2853         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2854         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
2855         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
2856         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
2857         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
2858         sourcename .. ".log " .. sourcename .. "_html.log " ..
2859         sourcename .. ".gl* " .. sourcename .. "_html.gl* " ..
```



```

2860         "*_html_inc.*"
2861     )
2862 end
2863
2864
2865
2866 -- Create lateximages based on lateximages.txt:
2867 function createlateximages ()
2868 print ("lwarpmk: Creating lateximages.")
2869 local limagesfile = io.open("lateximages.txt")
2870 -- Create the lateximages directory, ignore error if already exists
2871 err = os.execute("mkdir lateximages")
2872 -- Scan lateximages.txt
2873 for line in limagesfile:lines() do
2874 -- lwimgpage is the page number in the PDF which has the image
2875 -- lwimgnum is the sequential lateximage number to assign for the image
2876 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
2877 -- For each entry:
2878 if ( i~=nil ) then
2879 -- Separate out the image into its own single-page pdf:
2880 err = os.execute(
2881 "pdfseparate -f " .. lwimgpage .. " -l " ..
2882 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2883 -- Crop the image:
2884 err = os.execute(
2885 "pdfcrop --hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2886 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2887 -- Convert the image to svg:
2888 err = os.execute(
2889 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2890 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2891 -- Move the result into lateximages/:
2892 err = os.execute(
2893 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirsslash )
2894 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2895 -- Remove the temporary files:
2896 err = os.execute(
2897 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2898 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2899 end
2900 end -- do
2901 io.close(limagesfile)
2902 end -- function
2903
2904
2905 -- Use latexmk to compile source and index:
2906 -- fsuffix is "" for print, or "_html" for HTML
2907 function compilelatexmk ( fsuffix )
2908     -- The recorder option is required to detect changes in <project>.tex
2909     -- while we are loading <project>_html.tex.

```

```

2910     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2911         .. "-e "
2912         .. opquote
2913         .. "$makeindex = q/" -- $
2914         .. xindycmd
2915         .. " -M " .. xdyfile
2916         .. " -L " .. language .. " /"
2917         .. opquote
2918         .. " -pdflatex=\"\" .. latexname .. " %O %S\" "
2919         .. sourcename..fsuffix ..".tex" ) ;
2920     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2921 end
2922
2923
2924
2925 -- lwarpmk --version :
2926
2927 if (arg[1] == "--version") then
2928     print ( "lwarpmk: " .. printversion )
2929
2930 else -- not --version
2931
2932 -- print intro:
2933
2934 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
2935
2936 -- lwarpmk print:
2937
2938 if arg[1] == "print" then
2939     loadconf ()
2940     if ( latexmk == "true" ) then
2941         compilelatexmk ("" )
2942         print ("lwarpmk: Done.")
2943     else -- not latexmk
2944         verifyfileexists (sourcename .. ".tex") ;
2945         -- See if up to date:
2946         if (
2947             ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2948             (
2949                 lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2950                 lfs.attributes ( sourcename .. ".pdf" , "modification" )
2951             )
2952         ) then
2953             -- Recompile if not yet up to date:
2954             manytimes("")
2955             print ("lwarpmk: Done." ) ;
2956         else
2957             print ("lwarpmk: " .. sourcename .. ".pdf is up to date." ) ;
2958         end
2959     end -- not latexmk

```

```
2960
2961 elseif arg[1] == "print1" then
2962     loadconf ()
2963     verifyfileexists (sourcename .. ".tex") ;
2964     onetime("")
2965     print ("lwarpmk: Done.") ;
2966
2967 -- lwarp printindex:
2968 -- Compile the index then touch the source
2969 -- to trigger a recompile of the document:
2970
2971 elseif arg[1] == "printindex" then
2972 loadconf ()
2973 print ("lwarpmk: Processing the index.")
2974 os.execute(
2975     xindycmd
2976     .. " -M " .. xdyfile
2977     .. " -L " .. language
2978     .. " " .. sourcename .. ".idx")
2979 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2980 refreshdate ()
2981 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2982 print ("lwarpmk: Done.")
2983
2984 -- lwarp printglossary:
2985 -- Compile the glossary then touch the source
2986 -- to trigger a recompile of the document:
2987
2988 elseif arg[1] == "printglossary" then
2989 loadconf ()
2990 print ("lwarpmk: Processing the glossary.")
2991
2992 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2993     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2994     .. sourcename .. ".glo")
2995 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2996 refreshdate ()
2997 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2998 print ("lwarpmk: Done.")
2999
3000 -- lwarpmk html:
3001
3002 elseif arg[1] == "html" then
3003 loadconf ()
3004 if ( latexmk == "true" ) then
3005     compilelatexmk ("_html")
3006     pdftohtml ()
3007     print ("lwarpmk: Done.")
3008 else -- not latexmk
3009     verifyfileexists ( sourcename .. ".tex" ) ;
```

```
3010 -- See if exists and is up to date:
3011 if (
3012   ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3013   (
3014     lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3015     lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3016   )
3017 ) then
3018   -- Recompile if not yet up to date:
3019   manytimes("_html")
3020   pdftohtml ()
3021   print ("lwarpmk: Done.")
3022 else
3023   print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3024 end
3025 end -- not latexmk
3026
3027 elseif arg[1] == "html1" then
3028   loadconf ()
3029   verifyfileexists ( sourcename .. ".tex" ) ;
3030   onetime("_html")
3031   pdftohtml ()
3032   print ("lwarpmk: Done.")
3033
3034 elseif arg[1] == "pdftohtml" then
3035   loadconf ()
3036   pdftohtml ()
3037
3038 -- lwarpmk htmlindex:
3039 -- Compile the index then touch the source
3040 -- to trigger a recompile of the document:
3041
3042 elseif arg[1] == "htmlindex" then
3043   loadconf ()
3044   print ("lwarpmk: Processing the index.")
3045   os.execute(
3046     xindycmd
3047     .. " -M " .. xdyfile
3048     .. " -L " .. language
3049     .. " " .. sourcename .. "_html.idx"
3050 )
3051   print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3052   refreshdate ()
3053   print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3054   print ("lwarpmk: Done.")
3055
3056 -- lwarpmk htmlglossary:
3057 -- Compile the glossary then touch the source
3058 -- to trigger a recompile of the document:
3059
```

```
3060 elseif arg[1] == "htmlglossary" then
3061 loadconf ()
3062 print ("lwarpmk: Processing the glossary.")
3063
3064 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3065     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3066     "_html.gls " .. sourcename .. "_html.glo")
3067
3068 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3069 refreshdate ()
3070 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3071 print ("lwarpmk: Done.")
3072
3073 -- lwarpmk limages:
3074 -- Scan the lateximages.txt file to create lateximages,
3075 -- then touch the source to trigger a recompile.
3076
3077 elseif arg[1] == "limages" then
3078 loadconf ()
3079 print ("lwarpmk: Processing images.")
3080 createlateximages ()
3081 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3082 refreshdate ()
3083 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3084 print ("lwarpmk: Done.")
3085
3086 -- lwarpmk again:
3087 -- Touch the source to trigger a recompile.
3088
3089 elseif arg[1] == "again" then
3090 loadconf ()
3091 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3092 refreshdate ()
3093 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3094 print ("lwarpmk: Done.")
3095
3096 -- lwarpmk clean:
3097 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3098
3099 elseif arg[1] == "clean" then
3100 loadconf ()
3101 removeaux ()
3102 print ("lwarpmk: Done.")
3103
3104 -- lwarpmk cleanall
3105 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3106 -- and also project.pdf, *.html
3107
3108 elseif arg[1] == "cleanall" then
3109 loadconf ()
```

```

3110 removeaux ()
3111 os.execute ( rmname .. " " ..
3112     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
3113     "*.html"
3114 )
3115 print ("lwarpmk: Done.")
3116
3117 -- lwarpmk with no argument :
3118
3119 elseif (arg[1] == nil) then
3120 printhelp ()
3121
3122 -- lwarpmk -h or lwarpmk --help :
3123
3124 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3125 printusage ()
3126
3127 else
3128 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
3129 printhelp ()
3130 end
3131
3132 end -- not --version
3133 \end{filecontents*}
3134 % \end{Verbatim}% for syntax highlighting

3135 \end{LWR@createlwarpmk}

```

31 Stacks

for HTML output: 3136 \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:

```
3137 \newcommand*{\LWR@depthnone}{-5}
```

all sectioning depths are deeper than LWR@depthfinished:

```
3138 \newcommand*{\LWR@depthfinished}{-4}
3139 \newcommand*{\LWR@depthpart}{-1}
3140 \newcommand*{\LWR@depthchapter}{0}
3141 \newcommand*{\LWR@depthsection}{1}
3142 \newcommand*{\LWR@depthsubsection}{2}
3143 \newcommand*{\LWR@depthsubsubsection}{3}
3144 \newcommand*{\LWR@depthparagraph}{4}
3145 \newcommand*{\LWR@depthsubparagraph}{5}
```

used by \itemize, \enumerate, \description:

```
3146 \newcommand*{\LWR@depthlist}{6}
```

used by \item:

```
3147 \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.

```
3148 \newcommand*{\LWR@closeone}{}% top of the stack
3149 \newcommand*{\LWR@closetwo}{}
3150 \newcommand*{\LWR@closethree}{}
3151 \newcommand*{\LWR@closefour}{}
3152 \newcommand*{\LWR@closefive}{}
3153 \newcommand*{\LWR@closesix}{}
3154 \newcommand*{\LWR@closeseven}{}
3155 \newcommand*{\LWR@closeeight}{}
3156 \newcommand*{\LWR@closenine}{}
3157 \newcommand*{\LWR@closeten}{}
3158 \newcommand*{\LWR@closeeleven}{}
3159 \newcommand*{\LWR@closetwelve}{}%
```

31.3 Closing depths

A stack to record the depth of each level:



Note that nested \LaTeX structures may push depths which are non-sequential.

Ex:

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

```
3160 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
3161 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
3162 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
3163 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
3164 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
3165 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
3166 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
3167 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
3168 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
3169 \newcommand*{\LWR@closedepthten}{\LWR@depthnone}
3170 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3171 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

31.4 Pushing and popping the stack

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

Pushes one return action and its \LaTeX depth onto the stacks.

```
3172 \NewDocumentCommand{\pushclose}{m m}
3173 {
3174 \let\LWR@closetwelve\LWR@closeeeleven
3175 \let\LWR@closeeeleven\LWR@closeten
3176 \let\LWR@closeten\LWR@closenine
3177 \let\LWR@closenine\LWR@closeeeight
3178 \let\LWR@closeeeight\LWR@closeseven
3179 \let\LWR@closeseven\LWR@closesix
3180 \let\LWR@closesix\LWR@closefive
3181 \let\LWR@closefive\LWR@closefour
```



```

3182 \let\LWR@closefour\LWR@closethree
3183 \let\LWR@closethree\LWR@closetwo
3184 \let\LWR@closetwo\LWR@closeone
3185 \let\LWR@closeone#1
3186 \let\LWR@closedepthtwelve\LWR@closedeptheleven
3187 \let\LWR@closedeptheleven\LWR@closedephten
3188 \let\LWR@closedephten\LWR@closedepthnine
3189 \let\LWR@closedepthnine\LWR@closedeptheight
3190 \let\LWR@closedeptheight\LWR@closedepthseven
3191 \let\LWR@closedepthseven\LWR@closedepthsix
3192 \let\LWR@closedepthsix\LWR@closedepthfive
3193 \let\LWR@closedepthfive\LWR@closedepthfour
3194 \let\LWR@closedepthfour\LWR@closedepththree
3195 \let\LWR@closedepththree\LWR@closedephtwo
3196 \let\LWR@closedephtwo\LWR@closedepthone
3197 \let\LWR@closedepthone#2
3198 }

```

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

```

3199 \newcommand*{\popclose}
3200 {
3201 \let\LWR@closeone\LWR@closetwo
3202 \let\LWR@closetwo\LWR@closethree
3203 \let\LWR@closethree\LWR@closefour
3204 \let\LWR@closefour\LWR@closefive
3205 \let\LWR@closefive\LWR@closesix
3206 \let\LWR@closesix\LWR@closeseven
3207 \let\LWR@closeseven\LWR@closeeight
3208 \let\LWR@closeeight\LWR@closenine
3209 \let\LWR@closenine\LWR@closeten
3210 \let\LWR@closeten\LWR@closeeleven
3211 \let\LWR@closeeleven\LWR@closetwelve
3212 \let\LWR@closedepthone\LWR@closedephtwo
3213 \let\LWR@closedephtwo\LWR@closedepththree
3214 \let\LWR@closedepththree\LWR@closedepthfour
3215 \let\LWR@closedepthfour\LWR@closedepthfive
3216 \let\LWR@closedepthfive\LWR@closedepthsix
3217 \let\LWR@closedepthsix\LWR@closedepthseven
3218 \let\LWR@closedepthseven\LWR@closedeptheight
3219 \let\LWR@closedeptheight\LWR@closedepthnine
3220 \let\LWR@closedepthnine\LWR@closedephten
3221 \let\LWR@closedephten\LWR@closedeptheleven
3222 \let\LWR@closedeptheleven\LWR@closedepthtwelve
3223 }

3224 \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: 3225 `\begin{warpHTML}`

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

```
3226 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
3227 \ifthenelse{\isempty{#3}}{%
3228 {\csdef{#1#2}{}}
3229 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3230 }
```

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

```
3231 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
3232 \end{warpHTML}
```

33 HTML entities

for HTML output: 3233 `\begin{warpHTML}`

HTML entites and HTML Unicode entities:

```
3234 \let\LWR@origampersand\&
```

```
\HTMLentity {<entitytag>}
```

```
3235 \newcommand*{\HTMLentity}[1]{%
3236 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3237 \begingroup%
3238 \LWR@FBcancel%
3239 \LWR@origampersand#1;%

```

```

3240 \endgroup
3241 % \LWR@traceinfo{HTMLentity done}%
3242 }

\HTMLunicode {\hex_unicode}

3243 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}}

\&

3244 \renewcommand*{\&}{\HTMLentity{amp}}

\textless
\textgreater
3245 \let\LWR@origtextless\textless
3246 \renewcommand*{\textless}{\HTMLentity{lt}}
3247
3248 \let\LWR@origtextgreater\textgreater
3249 \renewcommand*{\textgreater}{\HTMLentity{gt}}

3250 \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: 3251 \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.

```
3252 \providecommand*{\BaseJobname}{\jobname}
```

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

```
3253 \providecommand*{\HTMLFilename}{}
```

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

```
3254 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

`\SetHTMLFileNumber` $\{\langle number \rangle\}$

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.

```
3255 \newcommand*{\SetHTMLFileNumber}[1]{%
3256 \setcounter{LWR@htmlfilenumber}{#1}%
3257 }
```

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

Defaults to use section names in the filenames.

```
3258 \newbool{FileSectionNames}
3259 \booltrue{FileSectionNames}
```

```
3260 \end{warpall}
```

for HTML output: `3261 \begin{warpHTML}`

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

```
3262 \newcounter{LWR@htmlfilenumber}
3263 \setcounter{LWR@htmlfilenumber}{0}
```

`\LWR@htmlsectionfilename` $\{\langle htmlfilenumber \text{ or } name \rangle\}$

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

```
3264 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3265 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
3266 \LWR@traceinfo{about to assign temp}%
3267 \edef\LWR@tempone{#1}%
3268 \LWR@traceinfo{about to compare with ??}%
3269 \ifthenelse{\equal{\LWR@tempone}{??}}%
```

```

3270 {\LWR@traceinfo{found ??}}%
3271 {\LWR@traceinfo{not found ??}}%
3272 \LWR@traceinfo{about to compare with zero or empty}%
3273 \ifthenelse{%
3274     \equal{\LWR@tempone}{0}%
3275     \OR \equal{\LWR@tempone}{}%
3276     \OR \equal{\LWR@tempone}{??}}%
3277 }%
3278 {%
3279     \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3280     \HomeHTMLFilename.html%
3281 }%

```

For a H₁E₁X section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename `index.html`:

```

3282 {%
3283     \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
3284     \ifthenelse{%
3285         \equal{\HTMLFilename}{%} \AND
3286         \equal{\LWR@tempone}{Index} \OR
3287         \equal{\LWR@tempone}{index}}%
3288     }%
3289     {%
3290         \LWR@traceinfo{prefixing the index name with an underscore.}%
3291         \_#1.html%
3292     }%

```

Otherwise, create a filename with the chosen prefix:

```

3293     {\HTMLFilename#1.html}%
3294 }%
3295 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
3296 }

```

`\LWR@htmlrefsectionfilename` $\{ \langle label \rangle \}$

Prints the filename for the given label

```

3297 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3298 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
3299 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
3300 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
3301 }

3302 \end{warpHTML}

```

35 Homepage link

for HTML output: 3303 \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.

```
3304 \newcommand*{\LinkHome}{%
3305 \LWR@subhyperrefclass{%
3306 \HomeHTMLFilename.html}%
3307 {Home}{linkhome}%
3308 }
```

\LWR@topnavigation Creates a link to the homepage at the top of the page for use when the window is too narrow for the sideroc.

```
3309 \newcommand*{\LWR@topnavigation}{
3310 \LWR@htmlElementclassline{nav}{topnavigation}{\LinkHome}
3311 }
```

\LWR@botnavigation Creates a link to the homepage at the bottom of the page for use when the window is too narrow for the sideroc.

```
3312 \newcommand*{\LWR@botnavigation}{
3313 \LWR@htmlElementclassline{nav}{botnavigation}{\LinkHome}
3314 }
```

```
3315 \end{warpHTML}
```

36 \PrintStack diagnostic tool



Diagnostics tool: Prints the \LaTeX 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: 3316 \begin{warpHTML}

\PrintStack Prints the closedepth stack.

```
3317 \newcommand*{\PrintStack}{
3318 \LWR@startpars
3319 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
```

```

3320 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3321 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3322 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
3323 }

3324 \end{warpHTML}

```

37 Closing stack levels

for HTML output: 3325 \begin{warpHTML}

Close one nested level:

```

3326 \newcommand*{\LWR@closeoneprevious}{%
3327
3328 \LWR@closeone{}
3329
3330 \popclose{}
3331 }

```

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

```

3332 \newcommand*{\LWR@closeprevious}[1]{

```

Close any pending paragraph:

```

3333 \LWR@stoppars

```

Close anything nested deeper than the desired depth:

```

3334 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
3335 }

```

```

3336 \end{warpHTML}

```

38 PDF pages and styles

for HTML output: 3337 \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.

```
3338 \newcommand{\LWR@forcenewpage}{%  
3339 \LWR@stoppars\LWR@orignewpage\LWR@startpars%  
3340 }
```

`\pagestyle`, etc. are nullified for HTML output.

`\pagestyle` $\{\langle style \rangle\}$

```
3341 \renewcommand*\pagestyle[1]{}{}
```

`\thispagestyle` $\{\langle style \rangle\}$

```
3342 \renewcommand*\thispagestyle[1]{}{}
```

`\markboth` $\{\langle left \rangle\} \{\langle right \rangle\}$

```
3343 \renewcommand*\markboth[2]{}{}
```

`\markright` $\{\langle right \rangle\}$

```
3344 \renewcommand*\markright[1]{}{}
```

`\raggedbottom`

```
3345 \renewcommand*\raggedbottom{}{}
```

`\flushbottom`

```
3346 \renewcommand*\flushbottom{}{}
```

`\sloppy`

```
3347 \renewcommand*\sloppy{}{}
```

`\fussy`

```
3348 \renewcommand*\fussy{}{}
```

`\pagenumbering` $\{\langle commands \rangle\}$

```
3349 \renewcommand*\pagenumbering[1]{}{}
```



```
3350 \end{warpHTML}
```

39 HTML tags, spans, divs, elements

for HTML output: 3351 \begin{warpHTML}

39.1 Mapping \LaTeX Sections to HTML Sections

```
3352 \newcommand*{\LWR@tagpart}{h2}
3353 \newcommand*{\LWR@tagpartend}{/h2}
3354 \newcommand*{\LWR@tagchapter}{h3}
3355 \newcommand*{\LWR@tagchapterend}{/h3}
3356 \newcommand*{\LWR@tagsection}{h4}
3357 \newcommand*{\LWR@tagsectionend}{/h4}
3358 \newcommand*{\LWR@tagsubsection}{h5}
3359 \newcommand*{\LWR@tagsubsectionend}{/h5}
3360 \newcommand*{\LWR@tagsubsubsection}{h6}
3361 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3362 \newcommand*{\LWR@tagparagraph}{\span class="paragraph"{} }
3363 \newcommand*{\LWR@tagparagraphend}{/span}
3364 \newcommand*{\LWR@tagsubparagraph}{\span class="subparagraph"{} }
3365 \newcommand*{\LWR@tagsubparagraphend}{/span}
3366
3367 \newcommand*{\LWR@tagregularparagraph}{p}
```

39.2 Babel-French

Adjust babel-french for HTML spaces. So far, this only works for pdf \LaTeX and x \LaTeX .

(Based on original code by DANIEL FLIPO.)

```
3368 \providecommand*{\LWR@FBcancel}{ }
3369
3370 \AtBeginDocument{%
3371 \ifundefined{frenchbsetup}%
3372 {}%
3373 {%
3374     \frenchbsetup{FrenchFootnotes=false}%
3375 %
3376     \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
3377     \renewcommand*{\FBcolonspace}{%
```

```

3378      \begingroup%
3379      \LWR@FBcancel%
3380      \LWR@origampersand{} \nbsp;%
3381      \endgroup%
3382  }%
3383  \renewcommand*{\FBthinspace}{%
3384      \begingroup%
3385      \LWR@FBcancel%
3386      \LWR@origampersand\#x202f;% \,
3387      \endgroup%
3388  }%
3389  \renewcommand*{\FBguillspace}{%
3390      \begingroup%
3391      \LWR@FBcancel%
3392      \LWR@origampersand{} \nbsp;% ~, for \og xyz \fg{}
3393      \endgroup%
3394  }%
3395  \DeclareDocumentCommand{\FBmedkern}{-}{-}{%
3396      \begingroup%
3397      \LWR@FBcancel%
3398      \LWR@origampersand\#x202f;% \,
3399      \endgroup%
3400  }%
3401  \DeclareDocumentCommand{\FBthickkern}{-}{-}{%
3402      \begingroup%
3403      \LWR@FBcancel%
3404      \LWR@origampersand{} \nbsp;% ~
3405      \endgroup%
3406  }%
3407  \renewcommand*{-}{\HTMLentity{nbsp}}}% was overwritten by babel-french
3408  \ifFBunicode%
3409  \else%
3410      \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
3411      \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
3412  \fi%
3413 }%
3414 }

```

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.

```

3415 \newcommand*{\LWR@htmltagc}[1]{%
3416 {%
3417 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%

```

```

3418 \begingroup%
3419 \LWR@FBcancel%
3420 \ifmmode\else\protect\LWR@origttfamily\fi%
3421 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3422 \endgroup%
3423 }%
3424 }

```

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.

```

3425 \newcommand*{\LWR@nestspanitem}{%
3426 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
3427 \LWR@origitem%
3428 }
3429
3430 \newenvironment*{LWR@nestspan}
3431 {%
3432 \LWR@traceinfo{LWR@nestspan}%
3433 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
3434 {%
3435 \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3436 }%
3437 {% not in a lateximage
3438 \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3439 \addtocounter{LWR@spandepth}{1}%
3440 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
3441 \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3442 \renewcommand{\BlockClassSingle}[2]{##2}%
3443 \renewcommand{\LWR@forcenewpage}{}%
3444 \renewcommand{\LWR@itemizestart}{%
3445 \let\item\LWR@nestspanitem%
3446 }%
3447 \renewcommand{\LWR@itemizeend}{}%
3448 \renewcommand{\LWR@enumeratestart}{%
3449 \let\item\LWR@nestspanitem%
3450 }%
3451 \renewcommand{\LWR@enumerateend}{}%
3452 \renewcommand{\LWR@descriptionstart}{%
3453 \let\item\LWR@nestspanitem%
3454 }%
3455 \renewcommand{\LWR@descriptionend}{}%
3456 }% not in a lateximage
3457 }% starting env
3458 {% ending env

```

```

3459 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
3460 {}%
3461 {\addtocounter{LWR@spandepth}{-1}}%
3462 \LWR@traceinfo{LWR@nestspan: done}%
3463 }
3464
3465 \AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

`\LWR@htmlspan` `{\langle tag \rangle}{\langle text \rangle}`



`\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.

```

3466 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3467 \LWR@ensuredoingapar%
3468 \LWR@htmltagc{#1}%
3469 \begin{LWR@nestspan}%
3470 #2%
3471 \LWR@htmltagc{/#1}%
3472 \end{LWR@nestspan}%
3473 }

```

`\LWR@htmlspanclass` `[\langle style \rangle]{\langle class \rangle}{\langle text \rangle}`

```

3474 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3475 \LWR@traceinfo{LWR@htmlspanclass #2}%
3476 \LWR@ensuredoingapar%
3477 \LWR@subhtmlclass{span}[#1]{#2}%
3478 \begin{LWR@nestspan}%
3479 #3%
3480 \LWR@htmltagc{/span}%
3481 \LWR@traceinfo{LWR@htmlspanclass done}%
3482 \end{LWR@nestspan}%
3483 }

```

`\LWR@htmltag` `{\langle tag \rangle}`

Print an HTML tag: `<tag>`

```

3484 \newcommand*{\LWR@htmltagb}[1]{%
3485 \LWR@htmltagc{#1}%
3486 \endgroup%
3487 }
3488
3489 \newcommand*{\LWR@htmltag}{%
3490 \begingroup\catcode'\_ =12

```

```

3491 \LWR@htmltagb%
3492 }

```

39.4 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment

3493 \newcommand*{\LWR@htmlopencomment}{%
3494 {%
3495 % \LWR@traceinfo{\LWR@htmlopencomment}%
3496 \begingroup%
3497 \LWR@FBcancel%
3498 \ifmmode\else\protect\LWR@origttfamily\fi%
3499 \LWR@origtextless{}!{-}{-}%
3500 \endgroup%
3501 }%
3502 }
3503
3504 \newcommand*{\LWR@htmlclosecomment}{%
3505 {%
3506 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3507 \begingroup%
3508 \LWR@FBcancel%
3509 \ifmmode\else\protect\LWR@origttfamily\fi%
3510 {-}{-}\LWR@origtextgreater{}%
3511 \endgroup%
3512 }%
3513 }

```

```

\LWR@htmlcomment  {\comment)}

```

```

3514 \newcommand{\LWR@htmlcomment}[1]{%
3515 \LWR@htmlopencomment}%
3516 {%
3517 \LWR@origttfamily% break ligatures
3518 #1%
3519 }%
3520 \LWR@htmlclosecomment{}}

```

```

\LWR@htmlblockcomment  {\comment)}

```

```

3521 \newcommand{\LWR@htmlblockcommentb}[1]
3522 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3523
3524 \newcommand{\LWR@htmlblockcomment}
3525 {%
3526 \begingroup\catcode'\_ =12%
3527 \LWR@htmlblockcommentb%
3528 }

```

`\LWR@htmlblocktag` `{<tag>}` print a stand-alone HTML tag

```

3529 \newcommand*{\LWR@htmlblocktag}[1]{%
3530 \LWR@stoppars%
3531 \LWR@htmltag{#1}%
3532 \LWR@startpars%
3533 }

```

39.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```

3534 \NewDocumentCommand{\LWR@subhtmllementclass}{m o m}{%
3535 \LWR@traceinfo{\LWR@subhtmllementclass #1 #3}%
3536 \IfValueTF{#2}%
3537 {% option
3538   \ifthenelse{\equal{#2}{}}{%
3539     {\LWR@htmltag{#1 class="#3"}}% empty option
3540     {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
3541 }% option
3542 {\LWR@htmltag{#1 class="#3"}}% no option
3543 \LWR@traceinfo{\LWR@subhtmllementclass done}%
3544 }

```

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

```

3545 \NewDocumentCommand{\LWR@htmllementclass}{m o m}{%
3546 \LWR@stoppars%
3547 \LWR@subhtmllementclass{#1}[#2]{#3}%
3548 \LWR@startpars%
3549 }

```

```

\LWR@htmlelementclassend  {\langle element \rangle} {\langle class \rangle}

3550 \newcommand*{\LWR@htmlelementclassend}[2]{%
3551 \LWR@stoppars%
3552 \LWR@htmltag{/#1}%
3553 \ifbool{HTMLDebugComments}{%
3554     \LWR@htmlcomment{End of #1 ‘‘#2’’}%
3555 }{}%
3556 \LWR@startpars%
3557 }

\LWR@htmldivclass  [{\langle style \rangle}] {\langle class \rangle}

3558 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3559 \LWR@htmlelementclass{div}[#1]{#2}%
3560 }

\LWR@htmldivclassend  {\langle class \rangle}

3561 \newcommand*{\LWR@htmldivclassend}[1]{%
3562 \LWR@htmlelementclassend{div}{#1}%
3563 }

```

39.6 Single-line elements

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

```

\LWR@htmlelementclassline  {\langle element \rangle} [{\langle style \rangle}] {\langle class \rangle} {\langle text \rangle}

3564 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
3565 \LWR@stoppars
3566 \LWR@subhtmlelementclass{#1}[#2]{#3}%
3567 #4%
3568 \LWR@htmltag{/#1}
3569 \LWR@startpars
3570 }

```

39.7 HTML5 semantic elements

```

\LWR@htmlmlelement  {\langle element \rangle}

```

```

3571 \newcommand*\LWR@htmlelement}[1]{%
3572 \LWR@htmlblocktag{#1}
3573 }

```

`\LWR@htmlelementend` `{\langle element \rangle}`

```

3574 \newcommand*\LWR@htmlelementend}[1]{%
3575 \LWR@stoppars
3576 \LWR@htmltag{/#1}
3577 \LWR@startpars
3578 }
3579
3580 \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.

Nullified versions are provided for print mode.

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

Env `BlockClass` `[\langle style \rangle] {\langle class \rangle}` High-level interface for `<div>` classes.

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

for HTML output:

```

3581 \begin{warpHTML}
3582 \NewDocumentEnvironment{BlockClass}{o m}%
3583 {
3584 \LWR@htmldivclass[#1]{#2}
3585 }
3586 {
3587 \LWR@htmldivclassend{#2}
3588 }
3589 \end{warpHTML}

```

for PRINT output:

```

3590 \begin{warpprint}
3591 \NewDocumentEnvironment{BlockClass}{o m}{\}{}%
3592 \end{warpprint}

```

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

for HTML output:

```

3593 \begin{warpHTML}
3594 \newcommand*\BlockClassSingle[2]{%
3595 \LWR@htmlelementclassline{div}{#1}{#2}%

```



```
3596 }
3597 \end{warpHTML}
```

for PRINT output: 3598 \begin{warpprint}
 3599 \newcommand{\BlockClassSingle}[2]{#2}
 3600 \end{warpprint}

`\InlineClass` [`\style`] [`\class`] [`\text`] High-level interface for inline span classes.

for HTML output: 3601 \begin{warpHTML}
 3602 \NewDocumentCommand{\InlineClass}{o m +m}{%
 3603 \LWR@htmlspanclass[#1]{#2}{#3}%
 3604 }
 3605 \end{warpHTML}

for PRINT output: 3606 \begin{warpprint}
 3607 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
 3608 \end{warpprint}

39.9 Closing HTML tags

for HTML output: 3609 \begin{warpHTML}

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

```
3610 \newcommand*\LWR@printclosepart}
3611   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3612 \newcommand*\LWR@printclosechapter}
3613   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3614 \newcommand*\LWR@printclosesection}
3615   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3616 \newcommand*\LWR@printclosesubsection}
3617   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3618 \newcommand*\LWR@printclosesubsubsection}
3619   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3620 \newcommand*\LWR@printcloseparagraph}
3621   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3622 \newcommand*\LWR@printclosesubparagraph}
3623   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```
3624 \newcommand*\LWR@printcloselistitem}
3625   {\LWR@htmltag{/li}}
3626 \newcommand*\LWR@printclosedescitem}
```

```

3627     {\LWR@htmltag{/dd}}
3628 \newcommand*{\LWR@printcloseitemize}
3629     {\LWR@htmltag{/ul}}
3630 \newcommand*{\LWR@printcloseenumerate}
3631     {\LWR@htmltag{/ol}}
3632 \newcommand*{\LWR@printclosedescription}
3633     {\LWR@htmltag{/dl}}

3634 \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 \LaTeX paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output: 3635 `\begin{warpHTML}`

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

```

3636 \newcounter{LWR@spandepth}
3637 \setcounter{LWR@spandepth}{0}

```

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

```

3638 \newbool{LWR@doingstartpars}
3639 \boolfalse{LWR@doingstartpars}

```

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

```

3640 \newbool{LWR@doingapar}
3641 \global\boolfalse{LWR@doingapar}

```

`\LWR@ensuredoingapar` If are about to print something visible, and if allowed to start a new paragraph, ensure that are `LWR@doingapar`, so that paragraph tags are placed:

```

3642 \newcommand*{\LWR@ensuredoingapar}{%
3643 \ifbool{LWR@doingstartpars}%

```

```

3644 {\global\booltrue{LWR@doingapar}}%
3645 {}%
3646 }

```

\LWR@openparagraph

```

3647 \newcommand*{\LWR@openparagraph}
3648 {%

```

See if paragraph handling is enabled:

```

3649 \ifbool{LWR@doingstartpars}%
3650 {% handling pars

```

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

```

3651     \ifthenelse{%
3652         \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3653         \cnttest{\value{LWR@spandepth}}{>}{0}%
3654     }% 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 .

```

3655     {}% no nested par tags

```

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

```

3656     {% yes nest par tags
3657         \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

See if are nested inside an item list:

```

3658         \ifnumcomp{\@listdepth}{>}{0}%
3659         {%

```

If so, leave some horizontal room in the \LaTeX PDF output for list labels:

```

3660             \LWR@origspace{1in}%
3661             }%
3662         {}%

```

Now have started a paragraph.

```
3663      \global\booltrue{LWR@doingapar}%
```

At the end of each paragraph, generate closing tag and do regular /par stuff. (Attempting to use the everyhook cr hook for \LWR@closeparagraph does not work well.)

```
3664      \let\par\LWR@closeparagraph%
3665  }% end of yes nest par tags
3666 }% end of handling pars
3667 {}% not handling pars
3668 }
```

\LWR@closeparagraph

```
3669 \newcommand*{\LWR@closeparagraph}
3670 {%
```

See if paragraph handling is enabled:

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

If currently in paragraph mode:

```
3672 {% handling pars
```

See if already started a lateximage or a :

```
3673   \ifthenelse{%
3674     \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3675     \cnttest{\value{LWR@spandepth}}{>}{0}%
3676   }%
```

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

```
3677   {% no nested par tags
3678     \ifthenelse{%
3679       \cnttest{\value{LWR@spandepth}}{>}{0}%
3680       \AND%
3681       \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3682     }%
3683     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3684     {}%
3685   }% no nested par tags
```

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

```
3686    {% yes nest par tags
```

Print a closing tag and some extra vertical space:

```
3687        \unskip%
3688        \LWR@htmltagc{/LWR@tagregularparagraph}%
3689        \LWR@orignewline%
```

No longer doing a paragraph:

```
3690        \global\boolfalse{LWR@doingapar}%
3691 % Disable the special \env{minipage} \& \cs{hspace} interaction
3692 % until a new minipage is found:
3693 %    \begin{macrocode}
3694        \global\boolfalse{LWR@minipagethispar}%
3695    }% end of yes nest par tags
3696}% end of handling pars
```

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

```
3697 {% not handling pars
3698    \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}{%
3699        {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3700    }{%
3701}% not handling pars
```

Finish with regular paragraph processing

```
3702 \LWR@origpar%
3703 }
```

```
3704 \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: 3705 \begin{warpHTML}

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

```
3706 \newcommand*{\LWR@startpars}%
3707 {%
```

Ignore if inside a span:

```
3708 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3709 {}%
3710 {%
```

See if currently handling HTML paragraphs:

```
3711 \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3712 {}%
```

If not currently in paragraph mode:

```
3713 {%
```

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

```
3714 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3715 \let\par\LWR@closeparagraph
3716
3717 }% an intentionally blank line
```

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

```
3718 \global\setbool{LWR@doingstartpars}{true}%
```

No `<par>` tag yet to undo:

```
3719 \global\boolfalse{LWR@doingapar}%
3720 }% nestspan
3721 }
```

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

```
3722 \newcommand*{\LWR@stoppars}%
3723 {%
```

Ignore if inside a span:

```
3724 \ifthenelse{\cnttest{\value{\LWR@spandepth}}{>}{0}}%
3725 {}%
3726 {%
```

See if currently handling HTML paragraphs:

```
3727 \ifbool{\LWR@doingapar}%
```

if currently in an HTML paragraph:

```
3728 {%
```

Print a closing tag:

```
3729 \unskip%
3730 \LWR@htmltagc{/LWR@tagregularparagraph}%
3731 \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
3732 \global\boolfalse{\LWR@doingapar}%
```

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

```
3733 \global\boolfalse{\LWR@minipagethispar}
3734
3735 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
3736 {}%
```

See if currently allowing HTML paragraphs:

```
3737 \ifbool{\LWR@doingstartpars}%
```

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

```
3738 {\ClearPreHook{par}}%
```

Else: do nothing

```
3739 {}%
```

no longer in paragraph mode

```
3740 \global\setbool{LWR@doingstartpars}{false}%
```

no <p> tag to undo:

```
3741 \global\boolfalse{LWR@doingapar}%
```

```
3742 }% nestspan
```

```
3743 }
```

```
3744 \end{warpHTML}
```

42 Page headers and footers

for HTML & PRINT: 3745 \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.

```
3746 \newcommand{\LWR@firstpagetop}{} % for the home page alone
```

```
3747 \newcommand{\LWR@pagetop}{} % for all other pages
```

```
3748 \newcommand{\LWR@pagebottom}{}%
```

```
3749
```

```
3750 \newcommand{\LWR@setfirstpagetop}[1]{%
```

```
3751 \renewcommand{\LWR@firstpagetop}{#1}
```

```
3752 \catcode'\_ =8
```

```
3753 }
```

\HTMLFirstPageTop {<text and logos>}

```
3754 \newcommand{\HTMLFirstPageTop}{%
```

```
3755 \catcode'\_ =12
```

```
3756 \LWR@setfirstpagetop
```

```
3757 }
```

```
3758 \newcommand{\LWR@setpagetop}[1]{%
```

```
3759 \renewcommand{\LWR@pagetop}{#1}
```

```
3760 \catcode'\_ =8
```

```
3761 }
```

\HTMLPageTop {<text and logos>}

```
3762 \newcommand{\HTMLPageTop}{%
```

```
3763 \catcode'\_ =12
```



```

3764 \LWR@setpagetopb
3765 }

3766 \newcommand{\LWR@setpagebottomb}[1]{%
3767 \renewcommand{\LWR@pagebottom}{#1}
3768 \catcode'\_ =8
3769 }

```

`\HTMLPageBottom` `{\text and logos}`

```

3770 \newcommand{\HTMLPageBottom}{%
3771 \catcode'\_ =12
3772 \LWR@setpagebottomb
3773 }

3774 \end{warpall}

```

43 CSS

for HTML output: 3775 \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.

```
3776 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

```

3777 \newcommand*{\LWR@newcssb}[1]{%
3778 \renewcommand*{\LWR@currentcss}{#1}
3779 \catcode'\_ =8
3780 }
3781
3782 \newcommand*{\CSSFilename}{
3783 \catcode'\_ =12
3784 \LWR@newcssb
3785 }
3786 \end{warpHTML}

```

for PRINT output: 3787 \begin{warpprint}
3788 \newcommand*{\CSSFilename}[1]{
3789 \end{warpprint}

44 HTML meta description and author

for HTML & PRINT: 3790 \begin{warpall}

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

```
3791 \providecommand{\theauthor}{}
3792 \newcommand{\theHTMLAuthor}{\theauthor}
3793
3794 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

3795 \end{warpall}
```

for HTML & PRINT: 3796 \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.

```
3797 \newcommand{\LWR@currentHTMLDescription}{}
3798
3799 \newcommand{\HTMLDescription}[1]{%
3800 \renewcommand{\LWR@currentHTMLDescription}{#1}
3801 }
3802
3803 \end{warpall}
```

45 Footnotes

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



pfnote numbers

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

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

45.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the \LaTeX box `\LWR@footnotes`. Using this instead of the original `\footins` box avoids having footnotes be printed by the output routine, since footnotes should be printed per HTML page instead of per PDF page.

See section 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: 3804 `\begin{warpHTML}`

Patch \LaTeX footnotes to use a new `\box` for lwarp footnotes.

3805 `\newbox\LWR@footnotes`

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

`\@makefntext` `{\langle text \rangle}`

3806 `\long\def\@makefntext#1{\@thefnmark #1}`

`\@makefnmark`

3807 `\def\@makefnmark{\hbox{\@thefnmark}}`

Footnotes may be in regular text, in which case paragraphs are tagged, or in a table data cell or `lateximage`, in which case paragraph tags must be added manually.

In a `lateximage` during HTML output, the `lateximage` is placed inside a print-mode minipage, but the footnotes are broken out by:

```

\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
\let\@footnotetext\LWR@footnotetext

```

`\LWR@footnotetext` `{\text}`

```

3808 \long\def\LWR@footnotetext#1{%
3809 \global\setbox\LWR@footnotes=\vbox{%

```

Add to any current footnotes:

```

3810     \unvbox\LWR@footnotes%

```

Remember the footnote number for `\ref`:

```

3811     \protected@edef\@currentlabel{%
3812         \csname p@footnote\endcsname\@thefnmark%
3813     }% \@currentlabel

```

Open a group:

```

3814     \color@begingroup%

```

Use HTML superscripts even inside a `lateximage`:

```

3815     \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%

```

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

```

3816     \ifthenelse{%
3817         \boolean{LWR@doingstartpars} \AND%
3818         \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3819     }%
3820     {%}%
3821     {\LWR@htmltagc{\LWR@tagregularparagraph}}%

```

Append the footnote to the list:

```

3822     \@makefntext{#1}%

```

Closing paragraph tag:

```

3823     \ifthenelse{%
3824         \boolean{LWR@doingstartpars} \AND%
3825         \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3826     }%
3827     {\par}%

```

```

3828    {%
3829        \LWR@htmltagc{\LWR@tagregularparagraph}%
3830        \LWR@orignewline%
3831    }%

```

Close the group:

```

3832    \color@endgroup%
3833 }% vbox
3834 }%

```

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

```
3835 \let\@footnotetext\LWR@footnotetext
```

45.5 Minipage footnote implementation

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

```

3836 \long\def\@mpfootnotetext#1{%
3837 \global\setbox\@mpfootins\vbox{%
3838     \unvbox\@mpfootins
3839     \reset@font\footnotesize
3840     \hsize\columnwidth
3841     \@parboxrestore
3842     \protected@edef\@currentlabel
3843     {\csname p@mpfootnote\endcsname\@thefnmark}%
3844     \color@begingroup

```

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

```

3845     \ifthenelse{%
3846         \boolean{LWR@doingstartpars} \AND%
3847         \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
3848     {%
3849     }%
3850     {\LWR@htmltagc{\LWR@tagregularparagraph}}%

3851     \@makefntext{%
3852         \ignorespaces#1%
3853     }%

```

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

```
3854     \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

```

3855     {}%
3856     {%
3857         \LWR@htmltagc{/\LWR@tagregularparagraph}%
3858         \LWR@orignewline%
3859     }%
3860     \color@endgroup%
3861 }% vbox
3862 }

```

45.6 Printing pending footnotes

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

```

3863 \newcommand*{\LWR@printpendingfootnotes}{%
3864 \ifvoid\LWR@footnotes\else
3865     \LWR@forcenewpage
3866     \begin{BlockClass}{footnotes}
3867     \LWR@origmedskip
3868     \unvbox\LWR@footnotes
3869     \setbox\LWR@footnotes=\vbox{}
3870     \end{BlockClass}
3871 \fi
3872 }

```

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

```

3873 \newcommand*{\LWR@epubprintpendingfootnotes}{%
3874 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{%
3875 {\LWR@printpendingfootnotes}%
3876 }%
3877 }

3878 \end{warpHTML}

```

46 Marginpars

for HTML output: 3879 `\begin{warpHTML}`

`\marginpar` [*left*] [*right*]

```

3880 \renewcommand{\marginpar}[2] [] {%

```

```

3881 \LWR@htmlspanclass{marginpar}{#2}%
3882 }

3883 \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: 3884 \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.

```

3885 \newcounter{FileDepth}
3886 \setcounter{FileDepth}{-5}

```

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

```

3887 \newbool{CombineHigherDepths}
3888 \booltrue{CombineHigherDepths}

```

```

3889 \end{warpall}

```

for HTML output: 3890 \begin{warpHTML}

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

```

3891 \newcommand*{\LWR@thisfilename}{ }

```

\LWR@thisnewfilename The filename being sanitized.

```

3892 \newcommand*{\LWR@thisnewfilename}{ }

```

\LWR@filenamenoblanks {<filename>}

Convert blanks into dashes, removes short words, store result in \LWR@thisfilename.



Be sure that this does not result in filename collisions! Use the optional TOC caption entry parameter for formatting. Remember to `\protect` \TeX commands which appear in section names and TOC captions.

```
3893 \newcommand*{\LWR@filenamenoblanks}[1]{%
3894 \begingroup
```

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

```
3895 \LWR@nullfonts
3896 \renewcommand*{\LWR@htmltagc}[1]{%
```

Replaces common symbols and short words with hyphens:

```
3897 \edef\LWR@thisnewfilename{#1}
3898 \fullexpandarg
```

Convert spaces into hyphens:

```
3899 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctuation into hyphens:

```
3900 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3901 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
3902 \StrSubstitute{\LWR@thisnewfilename}%
3903 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
3904 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
3905 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3906 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
3907 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
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 \StrSubstitute{\LWR@thisnewfilename}%
3914 {\textless}{-}[\LWR@thisnewfilename]
3915 \StrSubstitute{\LWR@thisnewfilename}%
3916 {\textgreater}{-}[\LWR@thisnewfilename]
3917 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3918 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3919 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
3920 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
3921 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
3922 \StrSubstitute{\LWR@thisnewfilename}%
3923 {\textbackslash}{-}[\LWR@thisnewfilename]
3924 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
```



```

3925 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3926 %      "~{}" for babel
3927 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3928 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
3929 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]

```

Convert short words:

```

3930 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3931 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3932 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3933 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3934 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3935 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3936 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3937 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3938 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3939 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
3940 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
3941 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3942 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3943 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3944 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3945 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3946 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
3947 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

3948 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
3949 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3950 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3951 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3952 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3953 %      emdash
3954 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3955 %      endash
3956 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3957 \endgroup
3958 }

```

`\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.

```

3959 \newcommand*{\LWR@newhtmlfile}[1]{

```

At the bottom of the ending file:

```
3960 \LWR@htmlelementclassend{section}{textbody}
3961
3962 \LWR@printpendingfootnotes
3963
```

No footer between files if EPUB:

```
3964 \ifbool{FormatEpub}{
3965 {}
3966 {
3967     \LWR@htmlelement{footer}
3968
3969     \LWR@pagebottom
3970
3971     \LWR@htmlelementend{footer}
3972 }
```

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

```
3973 \ifthenelse{\boolean{FormatEpub}\OR\boolean{FormatWordProcessor}}{
3974 {}
3975 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
3976 \LWR@stoppars
3977 \LWR@htmltag{/body}\LWR@orignewline
3978 \LWR@htmltag{/html}\LWR@orignewline
3979 \LWR@orignewpage
3980
3981 \addtocounter{LWR@htmlfilenumber}{1}%
```

If using a filename, create a version without blanks. The filename without blanks will be placed into \LWR@thisfilename. If not using a filename, the file number will be used instead.

```
3982 \ifbool{FileSectionNames}%
3983 {\LWR@filenamenoblanks{#1}}
3984 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilenumber}}
```

Include an HTML comment to instruct lwarpmk where to split the files apart. Uses pipe-separated fields for split_html.gawk. Uses monospaced font with ligatures disabled for everything except the title.

```

3985 \LWR@htmlblockcomment{%
3986 |Start file|%
3987 \LWR@htmlsectionfilename{\LWR@thisfilename}|%
3988 }

```

At the top of the starting file:

```

3989 \LWR@stoppars
3990

3991 \LWR@filestart{ -- #1}% there is an Emdash in front of the #1
3992

```

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

```

3993 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3994 {}
3995 {\LWR@topnavigation}
3996

```

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

```

3997 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3998 {}
3999 {
4000   \LWR@htmlelement{header}
4001
4002   \LWR@pagetop
4003
4004   \LWR@htmlelementend{header}
4005 }
4006

```

Print title only if there is one. Skip if formatting for an EPUB or word processor:

```

4007 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4008 {}
4009 {\ifcvoid{thetitle}{}\{\LWR@printthetitle\}}
4010

```

No sidetoc if formatting for an EPUB or word processor:

```

4011 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4012 {}
4013 {\LWR@sidetoc}
4014

```

Start of the <textbody>:

```
4015 \LWR@htmlclass{section}{\textbody}
4016
```

Keep paragraph tags disabled for now:

```
4017 \LWR@stoppars
4018
```

Track the page numbers:

```
4019 \setcounter{LWR@latestautopage}{\value{page}}
4020 }

4021 \end{warpHTML}
```

48 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing \TeX 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 pdf\LaTeX to generate characters and pdf\LaTeX to read them. If extra symbols appear in the text, it may be that pdf\LaTeX is actually producing a symbol over or under a character, resulting in pdf\LaTeX picking up the accent symbol separately.



Xe\LaTeX and Lua\LaTeX directly support accented section and file names.

for HTML output: 4022 \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.

```
4023 \newbool{LWR@forcinghtmlpage}
4024 \boolfalse{LWR@forcinghtmlpage}
4025
4026 \newcommand*{\ForceHTMLPage}{%
```

```
4027 \global\booltrue{LWR@forcinghtmlpage}%
4028 }
```

\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.

```
4029 \newbool{LWR@forcinghtmltoc}
4030 \boolfalse{LWR@forcinghtmltoc}
4031
4032 \newcommand*{\ForceHTMLTOC}{%
4033 \global\booltrue{LWR@forcinghtmltoc}%
4034 }
```

```
4035 \end{warpHTML}
```


for PRINT output:

```
4036 \begin{warpprint}
4037 \newcommand*{\ForceHTMLPage}{}
4038 \newcommand*{\ForceHTMLTOC}{}
4039 \end{warpprint}
```

for HTML output:

```
4040 \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.

```
4041 \newbool{LWR@mainmatter}
4042 \DeclareDocumentCommand{\mainmatter}{}{}%
4043 \booltrue{LWR@mainmatter}%
4044 }
```

\frontmatter Declare the front matter section of the document, using arabic numbering for the internal numbering. Does not reset the page number.

```
4045 \DeclareDocumentCommand{\frontmatter}{}{}%
4046 \boolfalse{LWR@mainmatter}%
4047 }
```

\backmatter Declare the back matter section of the document. Does not reset the page number.

```

4048 \DeclareDocumentCommand{\backmatter}{-}{-}{%
4049 \boolfalse{LWR@mainmatter}
4050 }

```

48.3 Sectioning support macros

`\LWR@sectionnumber` $\{\langle section\ type\rangle\}$

Typeset a section number and its trailing space with CSS formatting:

```

4051 \newcommand*{\LWR@sectionnumber}[1]{%
4052 \InlineClass{sectionnumber}{#1}%
4053 }

```

`autosec` A tag used by the TOC and index.

`\LWR@createautosec` $\{\langle section\ type\rangle\}$


Create an autosection tag.

```

4054 \newcommand*{\LWR@createautosec}[1]{%
4055 \LWR@htmltag{#1 id="autosec-\thepage"}{}}%
4056 }

```

`\LWR@pushoneclose` $\{\langle depth\rangle\}$ $\{\langle printclose\rangle\}$ 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.


```

4057 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}

```

`\LWR@startnewdepth` $\{\langle depth\rangle\}$ $\{\langle printclose\rangle\}$

Closes currently stacked tags of a lesser level, then opens the new nesting level by saving this new sectioning level's closing tag for later use.

 `\LWR@stoppars` must be executed first.

```

4058 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%

```

Close any stacked sections up to this new one.

```

4059 \LWR@closeprevious{#1}%

```

Push a new section depth:

```
4060 \LWR@pushoneclose{#1}{#2}%
4061 }
```

Ctrl 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.

```
4062 \newcounter{LWR@prevFileDepth}
4063 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

\@secntformat {<sectiontype>}

```
4064 \def\@secntformat#1{\csname the#1\endcsname\protect\quad}
```

\simplechapterdelim Used by tocibind and anonchap.

```
4065 \newcommand*{\simplechapterdelim}{}%
```

\@chapcntformat {<sectiontype>}

\let to \@secntformat by default, but may be redefined by \simplechapter and \restorechapter from tocibind or anonchap.

```
4066 \let\@chapcntformat\@secntformat
```

\LWR@section * [<TOC name>] {<name>} {<sectiontype>}

The common actions for the high-level sectioning commands.

```
4067 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
4068 \LWR@traceinfo{LWR@section |#2| |#3|}%
4069 % \ifthenelse{%
4070 % % \equal{#2}{\NoValue-}\AND%
4071 % % \isempty{#3}%
4072 % }%
4073 % {% empty section name
4074 % \LWR@traceinfo{LWR@section: empty section}%
4075 % }% empty section name
4076 % {% not an empty section name
4077 \LWR@traceinfo{LWR@section: not an empty section}%
4078 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
4079 \global\boolfalse{LWR@minipagethispar}%
```

Start a new HTML file unless starred, and if is a shallow sectioning depth.

Exception: Also start a new HTML file for \part*, for appendix.

Generate a new \LaTeX page so that toc and index page number points to the section:

```
4080 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4081 \IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{}%
4082 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
4083 \ifthenelse{%
4084   \(%
4085     \(\NOT\equal{#1}{\BooleanTrue})\)\OR%
4086     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart})\)\OR%
4087     \(\boolean{LWR@forcinghtmlpage})\)%
4088   \)%
4089   \AND%
4090   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
4091   \AND%
4092   \(%
4093     \NOT\boolean{CombineHigherDepths}\OR%
4094     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
4095   \)%
4096   \AND%
4097   \NOT\equal{#3}{}}% phantomsection
4098 }%
```

If so: start a new HTML file:

```
4099 {% new file
4100   \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
4101   \IfNoValueTF{#2}%
```

If no optional entry

```
4102   {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
4103   {\LWR@newhtmlfile{#2}}%
4104 }% new file
```


Else: No new HTML file:

```
4105 {% not new file
```

Generate a new \LaTeX page so that toc and index page number points to the section:

```
4106 \LWR@traceinfo{LWR@section: not a new HTML file}%
4107 \LWR@orignewpage%
4108
4109}% not new file
```

Remember this section's name for \nameref:

```
4110 \IfValueT{#3}{%
4111 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4112 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4113}%
```

Print an opening comment with the level and the name; ex: “section” “Introduction”

```
4114
4115 \ifbool{HTMLDebugComments}{%
4116 \LWR@htmlcomment{Opening #4 ‘‘#3’’}}%
4117 }{}%
4118
```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
4119 \ifthenelse{%
4120 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4121 }%
4122 {\LWR@startpars}%
4123 }%
```

Create the opening tag with an autosec:

```
4124 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

Check if starred:

```
4125 \IfBooleanTF{#1}%
4126 {%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
4127 \ifbool{LWR@forcinghtmltoc}%
4128 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4129 }
```

```
4129 {}%
4130}% starred
```

Not starred, so step counter and add to TOC:

```
4131}% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
4132    \ifthenelse{%
4133        \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4134    }%
4135    {% if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For article class, lwarp assumes that all is mainmatter.

```
4136        \LWR@traceinfo{LWR@section: about to test main matter}%
4137        \ifbool{LWR@mainmatter}%
4138        {%
4139            \LWR@traceinfo{LWR@section: yes mainmatter}%
4140            \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
4141        \LWR@traceinfo{LWR@section: about to addcontentsline}%
4142        \addcontentsline{toc}{#4}%
4143        {%
4144            \protect\numberline{\csuse{the#4}}%
4145            {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4146        }%
4147        \LWR@traceinfo{LWR@section: finished addcontentsline}%
4148   }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
4149        {% not main matter
4150            \LWR@traceinfo{LWR@section: no main matter}%
4151            \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4152        }% end of not main matter
4153   }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
4154    {%
4155        \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4156    }%
```

For part, print the section type:

```

4157 \ifbool{LWR@mainmatter}%
4158 {%
4159     \ifthenelse{%
4160         \(\cnttest{\csuse{LWR@depth#4}}{<=}%
4161         {\value{secnumdepth}}\)\ AND%
4162         \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}}\)%
4163     }%
4164     {\csuse{#4name}~{}}%
4165     {}%

```

Print the section number:

```

4166 \LWR@traceinfo{LWR@section: about to print section number}%
4167 \ifthenelse{%
4168     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4169 }%
4170 {%
4171     \ifstrequal{#4}{chapter}%
4172     {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
4173     {\protect\LWR@sectionnumber{\@secCNTformat{#4}}}%
4174 }%
4175 {}%
4176 \LWR@traceinfo{LWR@section: finished print section number}%
4177 }{}%
4178 }% end of not starred

```

Print the section name:

```

4179 #3%

```

Close the heading tag, such as /H2:

```

4180 \LWR@htmltag{\csuse{LWR@tag#4end}}%

```

Generate a \LaTeX label:

```

4181 \label{autopage-\thepage}%

```

Start paragraph handling unless is an inline paragraph or subparagraph:

```

4182 \ifthenelse{%
4183     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
4184 }%
4185 {\LWR@startpars}%
4186 {}%

```

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.

```

4187 \ifthenelse{%
4188     \NOT\equal{#1}{\BooleanTrue}\OR%
4189     \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
4190 }%
4191 {% not starred
4192     \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4193 }% not starred
4194 {}%
```

Reset to defaults if not a phantomsection:

```

4195 \ifstrempy{#3}%
4196 {}%
4197 {%
4198 \global\boolfalse{LWR@forcinghtmlpage}%
4199 \global\boolfalse{LWR@forcinghtmltoc}%
4200 }%
4201 %
4202 % }% not an empty section name
4203 \LWR@traceinfo{LWR@section: done}%
4204 }
```

48.4 `\section` and friends

`\part` * [*TOC name*] {*name*}

```

4205 \DeclareDocumentCommand{\part}{s o m}{%
4206 \LWR@epubprintpendingfootnotes%
4207 \LWR@stoppars%
4208
4209 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4210
4211 \LWR@section{#1}{#2}{#3}{part}%
4212 }
```

`\chapter` * [*TOC name*] {*name*}

```

4213 \let\@printcites\relax% for quotchap package
4214
4215 \@ifundefined{chapter}
4216 {}
4217 {%
4218 \DeclareDocumentCommand{\chapter}{s o m}{%
4219 \LWR@traceinfo{chapter #3}%
4220 \LWR@epubprintpendingfootnotes%
4221 \LWR@stoppars%
4222
4223 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4224
4225 \LWR@section{#1}{#2}{#3}{chapter}%
4226 \@printcites% for quotchap package
4227 }
4228 }

```

`\section * [⟨TOC name⟩] {⟨name⟩}`

```

4229 \DeclareDocumentCommand{\section}{s o m}{%
4230 \LWR@epubprintpendingfootnotes%
4231 \LWR@stoppars%
4232
4233 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4234
4235 \LWR@section{#1}{#2}{#3}{section}%
4236 }

```

`\subsection * [⟨TOC name⟩] {⟨name⟩}`

```

4237 \DeclareDocumentCommand{\subsection}{s o m}{%
4238 \LWR@epubprintpendingfootnotes%
4239 \LWR@stoppars%
4240
4241 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4242
4243 \LWR@section{#1}{#2}{#3}{subsection}%
4244 }

```

`\subsubsection * [⟨TOC name⟩] {⟨name⟩}`

```

4245 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4246 \LWR@epubprintpendingfootnotes%
4247 \LWR@stoppars%
4248
4249 \LWR@startnewdepth{\LWR@depthsubsubsection}%
4250 {\LWR@printclosesubsubsection}%
4251

```

```
4252 \LWR@section{#1}{#2}{#3}{subsubsection}%
4253 }
```

`\paragraph` * [*TOC name*] {*name*}

```
4254 \DeclareDocumentCommand{\paragraph}{s o m}{%
4255 \LWR@epubprintpendingfootnotes%
4256 \LWR@stoppars%
4257
4258 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4259
4260 \LWR@section{#1}{#2}{#3}{paragraph}%
4261 }
```

`\subparagraph` * [*TOC name*] {*name*}

```
4262 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4263 \LWR@epubprintpendingfootnotes%
4264 \LWR@stoppars%
4265
4266 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4267
4268 \LWR@section{#1}{#2}{#3}{subparagraph}%
4269 }

4270 \end{warpHTML}
```

49 Starting a new file

for HTML & PRINT: 4271 \begin{warpall}

`\HTMLLanguage` Default language for the HTML lang tag.

```
4272 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4273
4274 \newcommand*{\HTMLLanguage}[1]{%
4275 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4276 }

4277 \end{warpall}
```

for HTML output: 4278 \begin{warpHTML}

`\LWR@filestart {\title_suffix}`

Creates the opening HTML tags.

```
4279 \newcommand*{\LWR@filestart}[1]{
4280 \LWR@traceinfo{\LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
4281 \begingroup
4282 \LWR@traceinfo{\LWR@filestart: A}
4283 \LWR@nullfonts
4284 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
4285 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4286 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
4287 \LWR@htmltag{html lang="\LWR@currentHTMLELanguage"}\LWR@orignewline
```

Start of the meta data:

```
4288 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4289 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

Author:

```
4290 \ifcempty{theHTMLAuthor}{-}{-}{
4291 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline
4292 }
```

lwarp is the generator:

```
4293 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /\LWR@orignewline
4294        \LWR@orignewline
```

If there is a description, add it now:

```
4295 \ifdefempty{\LWR@currentHTMLDescription}{-}{-}{
4296 \LWR@htmltag{%
4297 meta name="description" content="\LWR@currentHTMLDescription" /\LWR@orignewline
4298        \LWR@orignewline
```

4299 }%

Mobile-friendly viewport:

```
4300 \LWR@htmltag{meta name="viewport" %
4301 content="width=device-width, initial-scale=1.0" /}%
4302 \LWR@orignewline
```

IE patch:

```
4303 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
4304 \LWR@htmltag{%
4305 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{}%
4306 \LWR@htmltag{/script}\LWR@orignewline
4307 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4308 \ifcvoid{thetitle}{}{%
4309 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
4310 }%
```

The page's stylesheet:

```
4311 \LWR@htmltag{%
4312 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4313 \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.

```
4314 \ifbool{mathjax}%
4315 {%
4316 \boolfalse{LWR@verbtags}
4317 \verbatiminput{lwarp_mathjax.txt}%
4318 \booltrue{LWR@verbtags}
4319 \LWR@stoppars
4320 }% end of mathjax
4321 {}%
```

End of the header:

```
4322 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4323 \LWR@htmltag{body}\LWR@orignewline
```



```

4324 \endgroup
4325 \LWR@traceinfo{LWR@filestart: done}
4326 }

4327 \end{warpHTML}

```

50 Starting HTML output

for HTML output: 4328 \begin{warpHTML}

\LWR@LwarpStart Executed at the beginning of the entire document.

```

4329 \catcode'\$=\active
4330 \newcommand*{\LWR@LwarpStart}
4331 {%
4332 \LWR@traceinfo{LWR@lwarpStart}

```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```

4333 \ifbool{FormatWordProcessor}{%
4334   \setcounter{FileDepth}{-5}%
4335   \boolfalse{HTMLDebugComments}%
4336 }{}

```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```

4337 \edef\LWR@strresult{\HomeHTMLFilename}
4338 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4339 \edef\LWR@strresult{\HTMLFilename}
4340 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}

```

Force onecolumn and empty page style:

```

4341 \LWR@origonecolumn%
4342 \LWR@origpagestyle{empty}

```

Reduce chance of line overflow in verbatim environments:

```

4343 \LWR@origscriptsize%

```

In PDF output, don't allow line breaks to interfere with HTML tags:

```

4344 \LWR@origraggedright%
4345 \LetLtxMacro{\}\{\LWR@endoffline}%

```

Spread the lines for pdftotext to read them well:

```
4346 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4347 \setlength{\parindent}{0pt}
4348 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
4349 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
4350 \setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
4351 \renewcommand{\ps@plain}{}%
```

\backslash centering \backslash raggedleft \backslash raggedright	Not used in the HTML environment: <pre>4352 \renewcommand*{\centering}{} 4353 \renewcommand*{\raggedleft}{} 4354 \renewcommand*{\raggedright}{}%</pre>
---	---

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:

```
4355 \LetLtxMacro{\LWR@origtabular}{\tabular}
4356 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
4357 \LetLtxMacro{\tabular}{\LWR@tabular}
4358 \LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
4359 \let\LWR@origcaption\caption
```

Labels: \backslash ltx@label is used in amsmath environments and is also patched by cleveref.

Label in HTML

```
4360 \let\LWR@origltx@label\ltx@label
4361 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
4362 \let\TeX\LWR@TeX
4363 \let\LaTeX\LWR@LaTeX
4364 \let\LuaTeX\LWR@LuaTeX
4365 \let\LuaLaTeX\LWR@LuaLaTeX
4366 \let\XeTeX\LWR@XeTeX
4367 \let\XeLaTeX\LWR@XeLaTeX
4368 \let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
4369 \global\boolfalse{LWR@doingapar}
4370 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
4371 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4372 \LWR@filestart{}
4373 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4374 \LWR@htmltag{header}\LWR@orignewline
4375 \LWR@startpars
4376 \LWR@firstpagetop
4377 \LWR@stoppars
4378 \LWR@htmltag{/header}\LWR@orignewline
4379 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4380 \LWR@htmltag{section class="textbody"{{}}
```

Document and page settings:

```
4381 \mainmatter
4382 \LWR@origpagenumbering{arabic}
```

Patch the itemize, enumerate, and description environments and `\item`. This works with the native \TeX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
4383 \LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
4384 \catcode'\$=\active
```

Allow HTML paragraphs to begin:

```
4385 \LWR@startpars
4386 \LWR@traceinfo{LWR@lwarpStart: done}
4387 }
4388 \catcode'\$=3% math shift until lwarp starts
```

```
4389 \end{warpHTML}
```

51 Ending HTML output

for HTML output: 4390 \begin{warpHTML}

\LWR@requesttoc {<boolean>} {<suffix>} Requests that a toc, lof, or lot be generated.

```
4391 \newcommand*{\LWR@requesttoc}[2]{%
4392 \ifbool{#1}
4393 {
4394   \expandafter\newwrite\csuse{tf@#2}
4395   \immediate\openout \csuse{tf@#2} \jobname.#2\relax
4396 }{}
4397 }
```

\LWR@LwarpEnd Final stop of all HTML output:

```
4398 \newcommand*{\LWR@LwarpEnd}
4399 {
4400 \LWR@stoppars
4401 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
4402 \LWR@htmllementclassend{section}{textbody}
```

Print any pending footnotes:

```
4403 \LWR@printpendingfootnotes
```

Create the footer:

```
4404 \LWR@htmllement{footer}
4405
4406 \LWR@pagebottom
4407
4408 \LWR@htmllementend{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.

```
4409 \ifthenelse{\boolean{FormatEpub}}{\OR\boolean{FormatWordProcessor}}
4410 {}
4411 {
4412     \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
```

```
4413 }

4414 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4415 \LWR@htmltag{/body}\LWR@orignewline
4416 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4417 \LWR@orignewpage
```

For lateximage commands:

```
4418 \immediate\closeout\LWR@lateximagesfile
4419 }
```

```
4420 \end{warpHTML}
```

52 Title page

package support lwarp supports the native \TeX 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.

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 \TeX 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
- `\@author` `\thanks`, `\and`, or `\affiliation`. These are low-level macros intended to be used by
- `\@date` 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
- `\printauthor` 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 `\thedata` are available if titling has been loaded, and
- `\theauthor` are sanitized user-level versions from which have been removed the `\thanks` and
- `\thedata` `\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 \TeX 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:

```
4421 \begin{warppall}
4422 \newrobustcmd{\affiliation}[1]{}
4423 \end{warppall}
```

for PRINT output:

```
4424 \begin{warpprint}

4425 \AtBeginEnvironment{titlepage}{
4426 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
4427 }
4428
4429 \AtBeginDocument{
4430 \@ifpackageloaded{titling}{
4431 \AtBeginEnvironment{titlingpage}{
4432 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
4433 }
4434 }{}% titling loaded
4435 }% AtBeginDocument

4436 \end{warpprint}
```

for HTML output:

```
4437 \begin{warppHTML}
```

Env titlepage Sets up a <div> of class titlepage.

```

4438 \renewenvironment*{titlepage}
4439 {
4440 \renewrobustcmd{\affiliation}[1]{\ \ \InlineClass{affiliation}{##1}}
4441 \LWR@printpendingfootnotes
4442 \LWR@forcenewpage
4443 \BlockClass{titlepage}
4444 }
4445 {
4446 \endBlockClass
4447 \LWR@printpendingfootnotes
4448 }

4449 \end{warpHTML}

```

52.3 Printing the thanks

for HTML & PRINT: 4450 \begin{warpall}

\printthanks Forces the \thanks to be printed.

This is necessary in a titlingpage environment when \maketitle was not used.

```

4451 \newcommand*{\printthanks}{\@thanks}

4452 \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: 4453 \begin{warpHTML}

\printtitle

```

4454 \newcommand*{\printtitle}
4455 {
4456 \LWR@stoppars
4457 \LWR@htmltag{h1}%
4458 \@title%
4459 \LWR@htmltag{/h1}
4460 \LWR@startpars
4461 }

```


`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
4462 \newcommand*{\LWR@printthetitle}
4463 {
4464 \LWR@stoppars
4465 \LWR@htmltag{h1}%
4466 \thetitle%
4467 \LWR@htmltag{/h1}
4468 \LWR@startpars
4469 }
```

`\printauthor` HTML version.

```
4470 \newcommand*{\printauthor}{
```

The entire author block is contained in a <div> named author:

```
4471 \begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
4472 \renewcommand{\and}{%
4473 \end{BlockClass}
4474 \begin{BlockClass}{oneauthor}
4475 }
```

Individual authors are contained in a <div> named oneauthor:

```
4476 \begin{BlockClass}{oneauthor}
4477 \@author
4478 \end{BlockClass}
4479 \end{BlockClass}
4480 }
```

`\printdate`

```
4481 \newcommand*{\printdate}{%
4482 \begin{BlockClass}{titledate}
4483 \@date
4484 \end{BlockClass}
4485 }
```

```
4486 \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: 4487 \begin{warpprint}

\printtitle

4488 \newcommand*{\printtitle}{\Huge\@title}}

\printauthor Print mode.

4489 \newcommand*{\printauthor}

4490 {\large\begin{tabular}[t]{c}\@author\end{tabular}}}

\printdate

4491 \newcommand*{\printdate}{\small\textit{\@date}}}

4492 \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: 4493 \begin{warppHTML}

4494 \if@titlepage

4495 \newcommand{\LWR@setfootnoteseries}{%

4496 \renewcommand\thefootnote{\@arabic\c@footnote}%

4497 }

4498 \else

4499 \newcommand{\LWR@setfootnoteseries}{%

4500 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%

4501 }

4502 \fi

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
4503 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
4504 \LWR@setfootnoteseries%
4505 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

```
\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
4506 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4507 \textsuperscript{\@thefnmark} \LWR@orighspace{1in}
```

```
\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
4508 ##1%
4509 }%
4510 }
```

`\@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`.

```
4511 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4512 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
4513 **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
4514 \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.

```
4515 \renewcommand*{\maketitle}{%
```

An HTML titlepage <div> is used for all classes.

```
4516 \begin{titlepage}
```

Set up special patches:

```
4517 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4518 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4519 \@thanks
```

Close the HTML titlepage div and cleanup:

```
4520 \end{titlepage}
4521 \setcounter{footnote}{0}%
4522 \global\let\thanks\relax
4523 \global\let\maketitle\relax
4524 \global\let\@maketitle\relax
4525 \global\let\@thanks\@empty
4526 \global\let\@author\@empty
4527 \global\let\@date\@empty
4528 \global\let\@title\@empty
4529 \global\let\title\relax
4530 \global\let\author\relax
4531 \global\let\date\relax
4532 \global\let\and\relax
4533 }
```

`\@maketitle` HTML mode. Typesets the title, etc.:

```
4534 \DeclareDocumentCommand{\@maketitle}{-}{%
4535   \LWR@stoppars\LWR@htmltag{h1}
4536   \@title
4537   \LWR@htmltag{/h1}\LWR@startpars
4538   \begin{BlockClass}{author}
4539     \renewcommand{\and}{
4540       \end{BlockClass}
4541       \begin{BlockClass}{oneauthor}
4542     }
4543     \begin{BlockClass}{oneauthor}
4544       \@author
4545     \end{BlockClass}
4546   \end{BlockClass}
```

```

4547 \begin{BlockClass}{titledate}
4548 \@date
4549 \end{BlockClass}
4550 }

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```

4551 \newcommand*{\LWR@titlingmaketitle}{%

```

Keep pending footnotes out of the title block:

```

4552 \@thanks

```

Set up special patches:

```

4553 \LWR@maketitlesetup

```

Typeset the title, etc:

```

4554 \@maketitle

```

Immediately generate any `\thanks` footnotes:

```

4555 \@thanks
4556 }

```

```

4557 \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: `4558 \begin{warpall}`

`\AddSubtitlePublished` Adds `\published` and `\subtitle`, and related.

```

4559 \AfterPackage{titling}{
4560 \newcommand*{\AddSubtitlePublished}{%
4561
4562 \newcommand{\@published}{%
4563
4564 \newcommand{\published}[1]{\gdef\@published{##1}}
4565
4566 \renewcommand*{\maketitlehooka}{\printpublished}
4567
4568 \newcommand*{\printpublished}{%
4569 \warpprintonly{\begin{center}\unskip}%
4570 \begin{BlockClass}{published}%
4571 \warpprintonly{\large\itshape}%
4572 \@published%
4573 \end{BlockClass}%
4574 \warpprintonly{\end{center}}}%
4575 }
4576
4577 \newcommand{\@subtitle}{%
4578
4579 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}
4580
4581 \renewcommand*{\maketitlehookb}{\printsubtitle}
4582
4583 \newcommand*{\printsubtitle}{%
4584 \warpprintonly{\begin{center}\unskip}%
4585 \begin{BlockClass}{subtitle}%
4586 \warpprintonly{\Large\itshape}%
4587 \@subtitle%
4588 \end{BlockClass}%
4589 \warpprintonly{\end{center}}}%
4590 }
4591
4592 }% \AddSubtitlePublished
4593 }% AfterPackage

4594 \end{warppall}

```

53 Abstract

The following code replaces the \LaTeX default, and will itself be replaced later if the abstract package is loaded.

for HTML output: 4595 \begin{warpHTML}

`\abstractname` User-redefinable title for the abstract.

Also over-written by the babel package.

```
4596 \providecommand*{\abstractname}{Abstract}
```

Env `abstract`

```
4597 \DeclareDocumentEnvironment{abstract}{}
4598 {
4599 \LWR@forcenewpage
4600 \BlockClass{abstract}
4601 \BlockClassSingle{abstracttitle}{\abstractname}
4602 }
4603 {
4604 \endBlockClass
4605 }

4606 \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:

```
4607 \begin{warpHTML}
4608 \newcommand{\attribution}[1]{%
4609 \InlineClass{attribution}{--\,#1}}% emdash
4610 \end{warpHTML}
```

for PRINT output:

```
4611 \begin{warpprint}
4612 \newcommand{\attribution}[1]{\textsc{---\,#1}}
4613 \end{warpprint}
```

`\citetitle` for use inside quote, quotation, verse:

for HTML output:

```
4614 \begin{warpHTML}
4615 \newcommand{\citetitle}[1]{%
4616 \InlineClass{citetitle}{--\,#1}}% emdash
4617 \end{warpHTML}
```

for PRINT output: 4618 \begin{warpprint}
 4619 \newcommand{\citetitle}[1]{\textsl{---\, #1}}
 4620 \end{warpprint}

54.2 Quotes, quotations

for HTML output: 4621 \begin{warpHTML}

Env quote

```
4622 \renewenvironment*{quote}
4623 {
4624 \LWR@forcenewpage
4625 \LWR@htmlblocktag{blockquote}
4626 }
4627 {\LWR@htmlblocktag{/blockquote}}
4628
4629 \renewenvironment*{quotation}
4630 {
4631 \LWR@forcenewpage
4632 \LWR@htmlblocktag{blockquotation}
4633 }
4634 {\LWR@htmlblocktag{/blockquotation}}
4635 \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: 4636 `\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 `\leftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

Len `\TMLvleftskip` Sets `\vleftskip` inside a verse environment in HTML.

```
4637 \newlength{\TMLvleftskip}
4638 \setlength{\TMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a verse environment in HTML.

```
4639 \newlength{\TMLleftmargini}
4640 \setlength{\TMLleftmargini}{4.5em}
```

```
4641 \end{warpall}
```

55 Verbatim

for HTML & PRINT: 4642 `\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.

```
4643 \newlength{\VerbatimHTMLWidth}
4644 \setlength{\VerbatimHTMLWidth}{4in}
4645 \end{warpall}
```

for HTML output: 4646 `\begin{warpHTML}`

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
4647 \newbool{LWR@verbtags}
4648 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [*<style>*] {*<class>*}

Encloses a verbatim environment with the given css class.

```
4649 \newcommand*{\LWR@atbeginverbatim}[2] []
4650 {%
```

Avoid excessive space between lines:

```
4651 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4652 \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.

```
4653 \ifbool{LWR@verbtags}{%
4654   \LWR@htmltag{pre class="#2"
4655   \ifthenelse{\equal{#1}{}}{}{}{style="#1"}}%
4656 }% pre
4657 }{}%
```

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:

```
4658 \begingroup%
4659 \LWR@origttfamily%
4660 \LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a verse par. Restore plain \TeX `\hspace` functionality:

```
4661 \LetLtxMacro{\hspace}{\LWR@orighspace}%
4662 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
4663 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the pre:

```
4664 \endgroup%
4665 \unskip%
```

At the end of the environment, close the pre:

```
4666 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
4667
4668 }{}%
```

Resume regular paragraph handling:

```
4669 \LWR@startpars%
4670 }
```

`\verbatiminput` $\{\langle filename \rangle\}$

Patch `\verbatiminput` to add HTML tags:

```
4671 \let\LWRV@origverbatim@input\verbatim@input
4672
4673 \renewcommand{\verbatim@input}[2]{%
4674 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
4675 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
4676 \LWRV@origverbatim@input{#1}{#2}%
4677 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4678 }
```

Env `verbatim`

```
4679 \AfterEndPreamble{
4680 \LWR@traceinfo{Patching verbatim.}
4681 \AtBeginEnvironment{verbatim}{%
4682 \LWR@forcenewpage
4683 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4684 }
4685 \AfterEndEnvironment{verbatim}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
4686 }

4687 \end{warpHTML}
```

56 Theorems

`\newtheorem` $\{\langle text \rangle\} [\langle counter \rangle] \text{-or-} [\langle oldname \rangle] \{\langle text \rangle\}$

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: 4688 `\begin{warpHTML}`

`\@begintheorem` $\{\langle name \rangle\} \{\langle number \rangle\}$

```
4689 \renewcommand{\@begintheorem}[2]{%
4690 \LWR@forcenewpage
4691 \BlockClass{theoremcontents}
4692 \InlineClass{theoremlabel}{#1\ #2\ }
4693 }
```

`\@opargbegintheorem` $\{\langle name \rangle\} \{\langle number \rangle\} \{\langle oparg \rangle\}$

```
4694 \renewcommand{\@opargbegintheorem}[3]{%
4695 \LWR@forcenewpage
4696 \BlockClass{theoremcontents}
4697 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4698 }
```

`\@endtheorem`

```
4699 \renewcommand*\@endtheorem{%
4700 \endBlockClass% theoremcontents
4701 }
```

```
4702 \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: 4703 `\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.

```
4704 \newcommand*{\LWR@itemizeitem}{%
4705 \LWR@stoppars%
4706 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitem}%
4707 \LWR@htmltag{li}%
4708 \LWR@startpars%
4709 \LWR@origitem%
4710 }
```

Env `itemize` [*<options>*]

```
4711 \newcommand*{\LWR@itemizestart}{%
4712 \LWR@stoppars%
4713 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize}%
4714 \LWR@htmltag{ul style="list-style-type:none"}%
4715 \LWR@startpars%
4716 \let\item\LWR@itemizeitem%
4717 }
4718
4719 \newcommand*{\LWR@itemizeend}{%
```

```

4720 \LWR@stoppars%
4721 \LWR@closeprevious{\LWR@depthlistitem}%
4722 \LWR@closeoneprevious{}%
4723 \LWR@startpars%
4724 }

```

57.2 Enumerate

An HTML unordered list is used with customized \LaTeX -generated labels.

Env `enumerate` [*(options)*]

```

4725 \newcommand*{\LWR@enumeratestart}{%
4726 \LWR@stoppars%
4727 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4728 \LWR@htmltag{ul style="list-style-type:none">{}}%
4729 \LWR@startpars%
4730 \let\item\LWR@itemizeitem%
4731 }
4732
4733 \newcommand*{\LWR@enumerateend}{%
4734 \LWR@stoppars%
4735 \LWR@closeprevious{\LWR@depthlistitem}%
4736 \LWR@closeoneprevious{}%
4737 \LWR@startpars%
4738 }

```

57.3 Description

`\LWR@descitem` [*(label)*] Handles an `\item` inside a description.

```

4739 \newcommand*{\LWR@descitem}[1][]{%
4740 {%
4741 \LWR@stoppars%
4742 \LWR@setlatestname{#1}%
4743 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printclosedescitem{}}%

```

Temporarily disable `\hspace`, which `article.cls`, etc. use per `\item` for descriptions only. This causes `lwarp` to mistakenly place an empty span between HTML list tags.

```

4744 \LetLtxMacro{\hspace}{\LWR@nohspace}%

```

Process the original `\item` code:

```
4745 \LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```
4746 \LetLtxMacro{\hspace}{\LWR@hspace}%
```

Be sure the label doesn't print to the left of the rest of the file:

```
4747 \LWR@orighspace{1in}
4748 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4749 \LWR@orignewline%
4750 \LWR@htmltag{dd}%
4751 \LWR@startpars%
4752 }
```

Env description [*options*]

```
4753 \newcommand*{\LWR@descriptionstart}{%
4754 \LWR@stoppars%
4755 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4756 \LWR@htmltag{dl}%
4757 \LWR@startpars%
4758 \let\item\LWR@descitem%
4759 }
4760
4761 \newcommand*{\LWR@descriptionend}{%
4762 \LWR@stoppars%
4763 \LWR@closeprevious{\LWR@depthlistitem}%
4764 \LWR@closeoneprevious{}}%
4765 \LWR@startpars%
4766 }
```

`\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 \TeX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```
4767 \newcommand*{\LWR@patchlists}{%
4768 \LetLtxMacro{\LWR@origitem}{\item}
4769 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4770 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4771 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4772 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
```

```

4773 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4774 \AtEndEnvironment{description}{\LWR@descriptionend}
4775 }

4776 \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 multirow, longtable, supertabular, or xtab.

Defining environments:

⚠ misplaced alignment
alignment tab character &

- When defining environments or macros which include tabular and instances of the & character, it may be necessary to make & active before the environment or macro is defined, then restore & to its default catcode after, using the following commands. These are 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}
{
\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.

```

... & \multirow{2}{.5in}{text} & ...
... & \mrowcell & ...

```

vposn

Note that recent versions of multirow include a new optional vposn argument.

- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warphTMLonly must be used to make versions for print and HTML.
- See section 174.2 for \multicolumnrow.

⚠ \multicolumn &
\multirow

lwarp does not support directly combining \multicolumn and \multirow. Use \multicolumnrow instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[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.

⚠ skipped cells

As per \multirow, skipped cells to the right of the \multicolumnrow statement are not included in the source code on the same line. On the following lines, \mcolrowcell must be used for each cell of each column and each row to be skipped:

⚠ empty cells

```

... & \multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...

```

vposn

Note that recent versions of multirow include a new optional vposn argument.

⚠ macro in a table
custom macros

- Using a custom macro inside a tabular data cell may result in an extra HTML data cell tag, corrupting the HTML table. To avoid this, use `\TabularMacro` just before the macro. This is ignored in print mode.

`\TabularMacro\some macro & more row contents \\`

Column specifiers:

⚠ * column specification

- * in a column specification is not used (so far). Repeat the column type the correct number of times.

@ and !

- Only one each of @ and ! is used at each column, and they are used in that order.

\multirow

- In `\multirow` cells, the print version may have extra instances of <, >, @, and ! cells on the second and later rows in the `\multirow` which do not appear in the HTML version.

⚠ \newcolumnntype

- `\newcolumnntype` is ignored; unknown column types are set to 1.

Rules:

vertical rules

- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.

width and trim

- Width options are honored. Trim options are converted to rounded top corners. Trim corners are not rounded with @ or ! columns, and full-width rules ignore trim.

full-width rules

- `\toprule`, `\midrule`, `\bottomrule`, and `\hline` ignore trim. When given an optional width, each cell is styled to create the custom border. Without an optional width, the entire row is given a class to assign the standard border.

combined rules

- If you wish to use `\cmidrule` followed by `\bottomrule`, it may be necessary to use:

```
\cmidrule{2-3} \\[-2ex]
\bottomrule
```

The optional `-2ex` is ignored in HTML but improves the visual formatting in the print output.

⚠ \warpprintonly
misplaced \noalign

- For `\toprule` and `\bottomrule`, when combined with a `warpprint` or `warppHTML` 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 \\}
\warppHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

58.2 Token lookahead

Used by `\LWR@futurenonpacelet` to look at the next token.

for HTML output: 4777 `\begin{warppHTML}`

`\LWR@mynexttoken`

4778 `\newcommand\LWR@mynexttoken\relax`

`\futurelet` copies the next token then executes a function to analyze

`\LWR@futurenonpacelet` does the same, but ignores intervening white space

Based on the booktabs style:

`\LWR@futurenonpacelet`

```
4779 \def\LWR@futurenonpacelet#1{\def\LWR@cs{#1}%
4780 \afterassignment\LWR@fnslone\let\nexttoken= }
4781 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
4782 \def\LWR@fnsltwo{%
4783 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
4784 \else\let\next=\nexttoken\fi\next}
4785 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

4786 `\newcommand*\LWR@getmynexttoken}{%`

```

4787 \LWR@traceinfo{LWR@getmynexttoken}%
4788 % nothing must follow this next line
4789 \LWR@futurenonpacelet\LWR@mynexttoken\LWR@tabledatacolumnntag
4790 }

```

58.3 Booleans

Bool LWR@startedrow True if should print a row tag before this column.

```

4791 \newbool{LWR@startedrow}
4792 \boolfalse{LWR@startedrow}

```

Bool LWR@doinghline True if the next row will have an hline or midrule above it. Also used for \midrule.

```

4793 \newbool{LWR@doinghline}
4794 \boolfalse{LWR@doinghline}

```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```

4795 \newbool{LWR@doingtbrule}
4796 \boolfalse{LWR@doingtbrule}

```

Bool LWR@doingcmidrule True if the next row will have a cmidrule above it.

This is used by \LWR@tabularfinishrow to force a final empty row to create the border for the \cmidrule.

```

4797 \newbool{LWR@doingcmidrule}
4798 \boolfalse{LWR@doingcmidrule}

```

Bool LWR@tableparcell True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```

4799 \newbool{LWR@tableparcell}

```

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

```

4800 \newbool{LWR@skippingmrowcell}

```

Bool LWR@skippingmcolrowcell True if are doing an empty \multicolumnrow cell, and thus there is no data tag to close, and do not print @ and ! columns.

```

4801 \newbool{LWR@skippingmcolrowcell}

```

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns

table data cells.

```
4802 \newbool{LWR@skipatbang}
```

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

```
4803 \newbool{LWR@emptyatbang}
```

Bool LWR@intabularmetadata True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
4804 \newbool{LWR@intabularmetadata}
```

```
4805 \boolfalse{LWR@intabularmetadata}
```

58.4 Handling &, @, !, and bar

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/>

[where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860](http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860)

\LWR@instertatbangcols

```
4806 \newcommand*{\LWR@insertatbangcols}{%
```

```
4807 \ifbool{LWR@skipatbang}%
```

```
4808 {}%
```

```
4809 {%
```

```
4810 \LWR@printatbang{at}{\theLWR@tablecolspos}%
```

```
4811 \LWR@printatbang{bang}{\theLWR@tablecolspos}%
```

```
4812 }%
```

```
4813 }
```

\LWR@closetabledatacell If LWR@skippingmrowcell or LWR@skippingmcolrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```
4814 \newcommand*{\LWR@closetabledatacell}{%
```

```
4815 \global\booltrue{LWR@intabularmetadata}%
```

```
4816 \ifbool{LWR@exitingtabular}{}%
```

```
4817 {% not exiting tabular
```

```
4818 \ifbool{LWR@skippingmrowcell or LWR@skippingmcolrowcell}%
```

```
4819 {%
```

If not skipping a \multicolumnrow cell, insert the @ and ! columns after this non-existent column.

```

4820      \ifbool{LWR@skippingmcolrowcell}%
4821      {}%
4822      {\LWR@insertatbangcols}%
4823  }%
4824  {% not skippingmrowcell

```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```

4825      \unskip%
4826      \ifboolexpr{%
4827          bool{LWR@tabularmutemods} or
4828          bool{LWR@skipatbang} or
4829          bool{LWR@emptyatbang}
4830      }%
4831      {}%
4832      {\LWR@getexparray{LWR@colafterspec}{\theLWR@tablecolspos}}%

```

Close paragraphs:

```

4833      \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
4834      \global\boolfalse{LWR@tableparcell}%

```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```

4835      \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
4836      \LWR@insertatbangcols%
4837  }% not skipping mrowcell
4838 }% not exiting tabular
4839 \global\boolfalse{LWR@skippingmrowcell}%
4840 \global\boolfalse{LWR@skippingmcolrowcell}%
4841 \global\boolfalse{LWR@skipatbang}%
4842 }

```

LWR@tabulardepth tracks whether & is being used inside a tabular.

```

4843 \newcounter{LWR@tabulardepth}
4844 \setcounter{LWR@tabulardepth}{0}
4845

```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```

4846 \let\LWR@origampmacro&

```

```

4847 \end{warpHTML}

```

58.4.1 Localizing & catcodes

for HTML & PRINT: 4848 \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.

```
4849 \newcommand{\StartDefiningTabulars}{%
4850 \LWR@traceinfo{StartDefiningTabulars}
4851 \warpHTMLonly{\catcode'\&=\active}%
4852 }
```

\EndDefiningTabulars Place after defining something with & in it.

```
4853 \newcommand{\EndDefiningTabulars}{%
4854 \LWR@traceinfo{EndDefiningTabulars}
4855 \warpHTMLonly{\catcode'\&=4}%
4856 }
```

```
4857 \end{warpall}
```

58.4.2 Handling &

for HTML output: 4858 \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.

```
4859 \newcommand*\LWR@tabularampersand{%
4860 \LWR@traceinfo{LWR@tabularampersand}%
4861 \ifthenelse{\cnttest{\value{LWR@tabulardepth}}{>}{0}}{
4862 }
```

If not skipping a multirow cell, close the current data cell.

```
4863 \unskip%
4864 \LWR@closetabledatacell%
```

Move to the next column.

```
4865 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
4866 \LWR@getmynexttoken%
4867 }%
```

If not inside a tabular, performs the original action:

```
4868 {\LWR@origampmacro}%
4869 }
```

& 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.

58.4.3 Filling an unfinished row

`\LWR@tabularfinishrow` Adds empty table cells if necessary to finish the row.

At the end of the table, if any bottom rules are requested then an empty row must be generated to form the borders which show the rules.

```
4870 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
4871 \ifboolexpr{%
4872   not bool {LWR@exitingtabular} or%
4873   bool{LWR@doingtbrule} or%
4874   bool{LWR@doingcmidrule} or%
4875   bool{LWR@doinghline} or%
4876   bool{LWR@startedrow}%
4877 }{%
```

To locally temporarily turn off `LWR@exitingtabular` so that table data tags will still be generated:

```
4878 \begingroup%
```


If generating a final row for the `\bottomrule` borders, turn off the `@`, `!`, `<`, and `>` column output:

```
4879 \ifbool{LWR@exitingtabular}{%
4880     \booltrue{LWR@tabularmutemods}%
4881 }{}%
```

Reenable the table data tags until finished with the final row:

```
4882 \boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
4883 \whiledo{\value{LWR@tablecolspos}<\value{LWR@tabletotalcols}}{%
4884 \LWR@tabulatedatasinglecolumn tag%
```

The following is essentially `\LWR@tabularampersand` with `LWR@emptyatbang` added to empty the following cells:

```
4885     \LWR@closetabledatacell%
4886     \addtocounter{LWR@tablecolspos}{1}%
4887     \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
4888     \LWR@getmynexttoken%
4889 }%
```

Required to close the final cell:

```
4890 \LWR@closetabledatacell%
```

Reenable the original `LWR@exitingtabular` to close the entire table:

```
4891 \endgroup%
4892 \global\boolfalse{LWR@emptyatbang}%
4893 }{}% ifboolexpr
4894 }
```

58.5 Handling `\\`

Inside `tabular`, `\\` is redefined to `\LWR@tabularendoffline`

Throws away options `\\[dim]` or `*`

\LWR@tabularendoffline

```

4895 \NewDocumentCommand{\LWR@tabularendoffline}{s o}
4896 {%
4897 \ifthenelse{\value{LWR@tablecolspos}<\value{LWR@tabletotalcols}}{%
4898   \LWR@tabularfinishrow%
4899 }%
4900 {%
4901   \LWR@closetabledatacell%
4902 }%
4903 %   \begin{macrocode}
4904 \LWR@htmltag{/tr}\LWR@orignewline
4905 \global\booltrue{LWR@intabularmetadata}

```

Not yet started a table row:

```

4906 \global\boolfalse{LWR@startedrow}

```

Additional setup:

```

4907 \global\boolfalse{LWR@doinghline}%
4908 \global\boolfalse{LWR@doingtbrule}%
4909 \global\boolfalse{LWR@doingcmidrule}%
4910 \LWR@clearmidrules%

```

Start at first column:

```

4911 \setcounter{LWR@tablecolspos}{1}

```

Look at the next token to decide between single column data tag or a special case:

```

4912 \LWR@getmynexttoken%
4913 }

```

58.6 Variables

```

4914 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4915 \newcommand*{\LWR@pposition}{}
4916 \newcommand*{\LWR@pleft}{}
4917 \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 bar.

```
4918 \newcommand*{\LWR@tablecolspec}{}
```

`\LWR@strresult` Holds the result of `Str` functions.

```
4919 \newcommand*{\LWR@strresult}{}
4920 \newcommand*{\LWR@strresulttwo}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
4921 \newcommand*{\LWR@origcolspec}{}

```

Ctr `LWR@tablecolwidth` 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.)

```
4922 \newcounter{LWR@tablecolwidth}

```

Ctr `LWR@tablecolspos` Where are currently looking into the table column specification. Index starts at 1.

```
4923 \newcounter{LWR@tablecolspos}

```

Ctr `LWR@tabletotalcols` Holds the final number of table columns, not counting `@` and `!` columns. This is equal to the number of cells in each row.

```
4924 \newcounter{LWR@tabletotalcols}

```

Ctr `LWR@tabletotalcolsnext` Holds the next column while parsing. Is one more than `LWR@tabletotalcols`.

```
4925 \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.

`LWR@colbarspec` A data array of specifications for vertical rules.

58.7 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
4926 \newcommand*{\LWR@colparameter}{}
```

`\LWR@parseatcolumn` Handles @{text} columns.

```
4927 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
4928 \LWR@traceinfo{at column}%
4929 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into `\LWR@strresult`, expanding once:

```
4930 \LWR@traceinfo{about to read the next token:}%
4931 \expandarg%
4932 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4933 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4934 \LWR@traceinfo{have now read the next token}%
4935 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
4936 {% left edge of the table:
4937   \LWR@traceinfo{at the left edge}%
4938   \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
4939   \LWR@traceinfo{at the left edge: %
4940   \LWR@getexparray{LWR@colatspec}{leftedge}}%
4941 }%
4942 {% not at the left edge:
4943   \LWR@traceinfo{not at the left edge}%
4944   \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4945   \LWR@traceinfo{at \theLWR@tabletotalcols: %
4946   \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}%
4947 }%
4948 \let\LWR@colparameter\relax%
4949 \booltrue{LWR@validtablecol}%
4950 }
```

`\LWR@parsebangcolumn` Handles !{text} columns.

```
4951 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
4952 \LWR@traceinfo{bang column}%
4953 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
4954 \LWR@traceinfo{about to read the next token:}%
4955 \expandarg%
4956 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4957 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4958 \LWR@traceinfo{have now read the next token}%
4959 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}{
4960 {% left edge of the table:
4961   \LWR@traceinfo{at the left edge}%
4962   \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
4963 }%
4964 {% not at the left edge:
4965   \LWR@traceinfo{not at the left edge}%
4966   \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4967   \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colparameter!}%
4968 }%
4969 \let\LWR@colparameter\relax%
4970 \booltrue{LWR@validtablecol}%
4971 }
```

\LWR@parsebeforecolumn Handles >{text} columns.

```
4972 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
4973 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into \LWR@strresult:

```
4974 \expandarg%
4975 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4976 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4977 \LWR@setexparray{LWR@colbeforesec}{\theLWR@tabletotalcolsnext}{\LWR@colparameter}%
4978 \let\LWR@colparameter\relax%
```

```
4979 \booltrue{LWR@validtablecol}%
4980 }
```

`\LWR@parseaftercolumn` Handles <{text} columns.

```
4981 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
4982 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```
4983 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@colparameter]
4984 \expandarg%
4985 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4986 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```
4987 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4988 \let\LWR@colparameter\relax%
4989 \booltrue{LWR@validtablecol}%
4990 }
```

`\LWR@parsebarcolumn` Handles vertical rules.

```
4991 \newcommand*{\LWR@parsebarcolumn}{%
4992 \LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
4993 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}%
4994 {% left edge of the table:
4995   \LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarl}%
4996 }%
4997 {% not at the left edge:
4998   \LWR@setexparray{LWR@colbarspec}{\theLWR@tabletotalcols}{tvertbarr}%
4999 }%
5000 \booltrue{LWR@validtablecol}%
5001 }
```

58.8 Parsing 'l', 'c', or 'r' columns

`\LWR@parsenormalcolumn` {<thiscolumn>}

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```

5002 \newcommand*{\LWR@parsenormalcolumn}[1]{%
5003 \appto\LWR@tablecolspec{#1}%
5004 \addtocounter{LWR@tablettotalcols}{1}%
5005 \addtocounter{LWR@tablettotalcolsnext}{1}%
5006 \LWR@traceinfo{normal column \theLWR@tablettotalcols: #1}%
5007 \LWR@setexparray{LWR@colatspec}{\theLWR@tablettotalcolsnext}{}%
5008 \LWR@setexparray{LWR@colbangspec}{\theLWR@tablettotalcolsnext}{}%
5009 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tablettotalcolsnext}{}%
5010 \LWR@setexparray{LWR@colafterspec}{\theLWR@tablettotalcolsnext}{}%
5011 \LWR@setexparray{LWR@colbarspec}{\theLWR@tablettotalcolsnext}{}%
5012 \booltrue{LWR@validtablecol}%
5013 }
```

58.9 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{\langle thiscolumn \rangle\}$ The width will be ignored.

```

5014 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```

5015 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```

5016 \addtocounter{LWR@tablecolspos}{1}%
5017 }
```

58.10 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{\langle thiscolumn \rangle\}$ The three parameters will be ignored.

```

5018 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```

5019 \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.

```
5020 \addtocounter{LWR@tablecolspos}{3}%
5021 }
```

58.11 Parsing the column specifications

⚠ HTML CSS cannot exactly match the \LaTeX concept of a baseline for a table row. Table 6 shows the \LaTeX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 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.

```
5022 \newbool{LWR@validtablecol}
```

\backslash LWR@parsetablecols $\{\langle colspecs \rangle\}$

Scans the column specification left to right.

Builds \backslash LWR@tablecolspec with the final specification, one column per entry. The final number of cells in each row is stored in LWR@tabletotalcols.

```
5023 \newcommand*{\LWR@parsetablecols}[1]{%
5024 \LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
5025 \renewcommand*{\LWR@origcolspec}{#1}%
```


Table 7: Tabular HTML column conversions

l, r, c:	Converted to table cells without paragraph tags. Uses css <code>vertical-align:middle</code> 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, \LaTeX places the top line of a parbox aligned with the rest of the text line, so css <code>vertical-align:bottom</code> 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 <code>td.tdp</code> , <code>td.tdP</code> , <code>td.tdprule</code> , and <code>td.tdPrule</code> to <code>vertical-align: middle</code> . Another possibility is to change L,R,C, and P to <code>vertical-align: top</code> and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
m:	With paragraph tags, css <code>vertical-align:middle</code> .
b:	With paragraph tags, css <code>vertical-align:top</code> 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:

```
5026 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
5027 \setcounter{LWR@tabletotalcols}{0}%
5028 \setcounter{LWR@tabletotalcolsnext}{1}%
5029 \LWR@setexparray{LWR@colatspec}{leftedge}{}%
5030 \LWR@setexparray{LWR@colatspec}{1}{}%
5031 \LWR@setexparray{LWR@colatspec}{2}{}%
5032 \LWR@setexparray{LWR@colatspec}{3}{}%
5033 \LWR@setexparray{LWR@colbangspec}{leftedge}{}%
5034 \LWR@setexparray{LWR@colbangspec}{1}{}%
5035 \LWR@setexparray{LWR@colbangspec}{2}{}%
5036 \LWR@setexparray{LWR@colbangspec}{3}{}%
5037 \LWR@setexparray{LWR@colbeforespec}{1}{}%
5038 \LWR@setexparray{LWR@colbeforespec}{2}{}%
5039 \LWR@setexparray{LWR@colbeforespec}{3}{}%
5040 \LWR@setexparray{LWR@colafterspec}{1}{}%
5041 \LWR@setexparray{LWR@colafterspec}{2}{}%
5042 \LWR@setexparray{LWR@colafterspec}{3}{}%
5043 \LWR@setexparray{LWR@colbarspec}{leftedge}{}%
5044 \LWR@setexparray{LWR@colbarspec}{1}{}%
5045 \LWR@setexparray{LWR@colbarspec}{2}{}%
5046 \LWR@setexparray{LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
5047 \setcounter{LWR@tablecolspos}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```
5048 \LWR@traceinfo{about to StrLen}%
5049 \noexpandarg%
5050 \StrLen{#1}[\LWR@strresult]%
5051 \fullexpandarg%
5052 \LWR@traceinfo{finished StrLen}%
5053 \setcounter{LWR@tablecolwidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
5054 \whiledo{\not\value{LWR@tablecolspos}>\value{LWR@tablecolwidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```

5055 \noexpandarg%
5056 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
5057 \LWR@traceinfo{position \arabic{LWR@tablecolspos}: \LWR@strresult}%
5058 \fullexpandarg%

```

Not yet found a valid column type

```

5059 \boolfalse{LWR@validtablecol}%

```



Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```

5060 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
5061 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
5062 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
5063 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
5064 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
5065 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
5066 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
5067 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
5068 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
5069 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
5070 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
5071 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
5072 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
5073 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
5074 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
5075 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

```

From the dcolumn package:

```

5076 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%

```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```

5077 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%

```

Many people define centered versions “P”, “M”, and “B”:

```

\newcolumntype{P}[1]{>\centering\arraybackslash}p{#1}}

```

```

5078 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
5079 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
5080 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to an l column:

```

5081 \ifbool{LWR@validtablecol}{}{%
5082   \LWR@traceinfo{invalid column type: \LWR@strresult}%
5083   \LWR@parsenormalcolumn{1}%
5084 }%
5085 \addtocounter{LWR@tablecolspos}{1}%
5086 }%
5087 }%

```

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.

```

5088 \newcommand*{\LWR@maybenewtablerow}
5089 {%
5090 \ifbool{LWR@startedrow}%
5091 {}% started the row
5092 {% not started the row

```

Remember that now have started the row:

```

5093   \global\booltrue{LWR@startedrow}%

```

Create the row tag, with a class if necessary.

```

5094   \global\booltrue{LWR@intabularmetadata}%
5095   \ifbool{LWR@doinghline}%
5096   {%
5097     \LWR@htmltag{tr class="hline" }%
5098     \LWR@orignewline%
5099   }%
5100   {% not doing hline
5101     \ifbool{LWR@doingtbrule}%
5102     {%
5103       \LWR@htmltag{tr class="tbrule"%}
5104       \LWR@orignewline%
5105     }%
5106     {\LWR@htmltag{tr}\LWR@orignewline}%
5107   }% end of not doing hline
5108 }% end of not started the row
5109 }

```

58.13 Printing vertical bar tags

`\LWR@printbartag` $\{\langle index \rangle\}$

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```

5110 \newcommand*{\LWR@printbartag}[1]{%
5111 \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}%
5112 {%
5113 }%
5114 { % not muting
5115     \ifstrequal{#1}{\leftedge}{%
5116         \LWR@getexparray{\LWR@colbarspec}{\leftedge} %
5117     }%
5118     {
5119         \LWR@getexparray{\LWR@colbarspec}{#1} %
5120         \edef\LWR@tempone{\LWR@getexparray{\LWR@colbarspec}{#1}}%
5121     } %
5122 }% not muting
5123 }
```

58.14 Printing at or bang tags

`\LWR@printatbang` $\{\langle at-or-bang \rangle\} \{\langle index \rangle\}$

```

5124 \newcommand*{\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```

5125 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@col#1spec}{#2}}%
5126 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```

5127 \ifdefempty{\LWR@atbangspec}%
5128 {}%
5129 {% not empty
5130     \LWR@htmltag{%
5131         td class="td#1%
5132         \LWR@subaddcmidruletrim{}{}%
5133         \LWR@printbartag{#2}%
5134         "%
5135         \LWR@addcmidrulewidth%
5136     }%
```

Create an empty cell if muting for the \bottomrule:

```

5137 \ifbool{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}{%
5138   }{%
5139   {\LWR@atbangspec}%
5140 %
5141   \LWR@htmltag{/td}\LWR@orignewline%
5142 }% not empty
5143 }%
```

\LWR@addleftmostbartag

```

5144 \newcommand*{\LWR@addleftmostbartag}{%
5145 \ifnumcomp{\value{LWR@tablecolspos}}{=}{1}{%
5146   \LWR@printbartag{leftedge}%
5147 }{}%
5148 }
```

\LWR@tabularleftedge

```

5149 \newcommand*{\LWR@tabularleftedge}{%
5150 \ifthenelse{\cnttest{\value{LWR@tablecolspos}}=1}{%
5151   {%
5152     \LWR@printatbang{at}{leftedge}%
5153     \LWR@printatbang{bang}{leftedge}%
5154 }% left edge
5155 }% not left edge
5156 }
```

58.15 Data opening tag

\LWR@tabledatasinglecolumnntag Print a table data opening tag with style for alignment

```

5157 \newcommand*{\LWR@tabledatasinglecolumnntag}{%
5158   {%
5159   \LWR@maybe newtable row%
```

Don't start a new paragraph tag if have already started one:

```

5160 \ifbool{LWR@intabularmetadata}{%
5161   {%
```

If have found the end of tabular command, do not create the next data cell:

```

5162     \ifbool{LWR@exitingtabular}{}%
5163     {% not exiting tabular

```

Print the @ and ! contents before first column:

```

5164         \LWR@tabularleftedge%

```

Fetch the current column's alignment character into \LWR@strresult:

```

5165         \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

5166         \LWR@htmltag{td class="td%

```

append this column's spec:

```

5167         \LWR@strresult%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag. Also add vertical bar tags.

```

5168         \LWR@addcmidruletrim%
5169         \LWR@addleftmostbartag%
5170         \LWR@printbartag{\theLWR@tablecolspos}%
5171         "%
5172         \LWR@addcmidrulewidth%
5173         }%

```

If this is a p, m, b, or X column, allow paragraphs:

```

5174         \ifthenelse{%
5175             \equal{\LWR@strresult}{p}\OR%
5176             \equal{\LWR@strresult}{m}\OR%
5177             \equal{\LWR@strresult}{b}\OR%
5178             \equal{\LWR@strresult}{P}\OR%
5179             \equal{\LWR@strresult}{M}\OR%
5180             \equal{\LWR@strresult}{B}\OR%
5181             \equal{\LWR@strresult}{X}%
5182         }%
5183         {% allow pars
5184             \LWR@startpars%
5185             \global\booltrue{LWR@tableparcell}%
5186         }% allow pars
5187         }% no pars

```

Print the > contents unless muted for the \bottomrule:

```

5188      \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}}%
5189      {}%
5190      {\LWR@getexpparray{LWR@colbeforespec}{\theLWR@tablecolspos}}}%
5191      \global\boolfalse{LWR@intabularmetadata}%
5192      }% not exiting tabular
5193 }{}% in tabular metadata
5194 }%

```

58.16 Midrules

LWR@midrules LWR@midrules is a data array (section 32) of columns each containing a non-zero width if a midrule should be created for this column.

LWR@trimlrules LWR@trimlrules is a data array (section 32) of columns containing 1 if a midrule should be left trimmed for each column.

LWR@trimrrules LWR@trimrrules is a data array (section 32) of columns containing r if a midrule should be right trimmed for each column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules and LWR@trim<l/r>rules data arrays.

```
5195 \newcounter{LWR@midrulecounter}
```

Len \LWR@heavyrulewidth The default width of the rule.

```
5196 \newlength{\LWR@heavyrulewidth}
5197 \setlength{\LWR@heavyrulewidth}{.08em}
```

Len \LWR@lightrulewidth The default width of the rule.

```
5198 \newlength{\LWR@lightrulewidth}
5199 \setlength{\LWR@lightrulewidth}{.05em}
```

Len \LWR@cmidrulewidth The default width of the rule.

```
5200 \newlength{\LWR@cmidrulewidth}
5201 \setlength{\LWR@cmidrulewidth}{.03em}
```

Len \LWR@thiscmidrulewidth The width of the next rule, defaulting to \lightrulewidth.

If not \lightrulewidth, a style will be used to generate the custom width.

Assigned from the LWR@midrules array.

```
5202 \newlength{\LWR@thiscmidrulewidth}
5203 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}
```


`\LWR@clearmidrules` Start new midrules. Called at beginning of tabular and also at `\.`

Clears all `LWR@midrules` and `LWR@trimrules` markers for this line.

```

5204 \newcommand*{\LWR@clearmidrules}
5205 {%
5206 \setcounter{LWR@midrulecounter}{1}%
5207 \whiledo{%
5208 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolwidth}}%
5209 }%
5210 {%
5211 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{0pt}%
5212 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5213 \LWR@setexparray{LWR@trimlrules}{\theLWR@midrulecounter}{}%
5214 \LWR@setexparray{LWR@trimrrules}{\theLWR@midrulecounter}{}%
5215 \addtocounter{LWR@midrulecounter}{1}%
5216 }%
5217 }

```

`\LWR@subcmidrule` `{<width>}{<trim>}{<leftcolumn>}{<rightcolumn>}`

Marks `LWR@midrules` data array elements to be non-zero widths from left to right columns. Also marks trimming for the L and/or R columns.

`LWR@doingcmidrule` is set to force an empty row at the end of the tabular to create the rule.

```

5218 \newcommand*{\LWR@subcmidrule}[4]{%
5219 \setcounter{LWR@midrulecounter}{#3}%
5220 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
5221 {%
5222 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{#1}%
5223 \addtocounter{LWR@midrulecounter}{1}%
5224 }% \whiledo
5225 \IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{1}}{}%
5226 \IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
5227 \booltrue{LWR@doingcmidrule}%
5228 }

```

`\LWR@docmidrule` `[<width>]{<trim>}{<leftcolumn-rightcolumn>}`

Marks `LWR@midrules` array elements to be “Y” from left to right columns. Also marks trimming for the L and/or R columns.

```

5229 \NewDocumentCommand{\LWR@docmidrule}{0{\LWR@cmidrulewidth} D(){} >\SplitArgument{1}{-}}m{%
5230 {\LWR@subcmidrule{#1}{#2}{#3}}

```

Used to compute margins, tabular trims

```
5231 \newlength{\LWR@templengthone}%
5232 \newlength{\LWR@templengthtwo}%
```

`\LWR@subaddcmidruletrim` $\{\langle lefttrim \rangle\} \{\langle righttrim \rangle\}$ Adds a `\cmidrule` with optional trim.

```
5233 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
5234 \setlength{\LWR@templengthone}{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}%
5235 \ifdimcomp{\LWR@templengthone}{>}{0pt}%
5236 {%
```

Print the class without left and right trim letters appended:

```
5237 \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
5238 \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
5239 }{}%
5240 }
```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```
5241 \newcommand*{\LWR@addcmidruletrim}{%
5242 \LWR@subaddcmidruletrim%
5243 {\LWR@getexparray{\LWR@trimlrules}{\theLWR@tablecolspos}}%
5244 {\LWR@getexparray{\LWR@trimrrules}{\theLWR@tablecolspos}}%
5245 }
```

`\LWR@addrulewidth` $\{\langle thiswidth \rangle\} \{\langle defaultwidth \rangle\}$

If not default width, add a custom style with width and color depending on `thiswidth`.

```
5246 \newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`:

```
5247 \ifdimcomp{#1}{=}{#2}%
5248 {}% default width
5249 {}% custom width
```

Ensure that the width is wide enough to display in the browser:

```
5250 \LWR@forceminwidth{#1}%
```

```

5251 \uselengthunit{PT} % explicit space
5252 style="border-top:\rndprintlength{\LWR@atleastonept} solid %

```

The darkness of the color depends on the thickness of the rule:

```

5253 \ifdimcomp{#1}{<}{\LWR@lightrulewidth}%
5254 {\#AOAOAO}%
5255 {% lightrule or heavier
5256 \ifdimcomp{#1}{<}{\LWR@heavyrulewidth}%
5257 {\#808080}%
5258 {black}%
5259 }% lightrule or heavier
5260 "%
5261 }% custom width
5262 }

```

`\LWR@addcmidrulewidth`

```

5263 \newcommand{\LWR@addcmidrulewidth}{%
5264 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5265 }

```

58.17 Multicolumns

58.17.1 Parsing multicolumns

```

5266 \newcounter{LWR@tablemulticolwidth}

```

Indexes into the multicolumn specification:

```

5267 \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.

```

5268 \newcommand*{\LWR@printmccoltype}[1]{%
5269 \LWR@traceinfo{lw@printmccoltype -#1-}%

```

Get one token of the column spec:

```

5270 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%

```

Add to the HTML tag depending on which column type is found:

```

5271 \IfStrEq{\LWR@strresult}{l}{l}{}%
5272 \IfStrEq{\LWR@strresult}{c}{c}{}%
5273 \IfStrEq{\LWR@strresult}{r}{r}{}%
5274 \IfStrEq{\LWR@strresult}{p}{p}{}%
5275 \IfStrEq{\LWR@strresult}{m}{m}{}%
5276 \IfStrEq{\LWR@strresult}{b}{b}{}%
5277 \IfStrEq{\LWR@strresult}{P}{P}{}%
5278 \IfStrEq{\LWR@strresult}{M}{M}{}%
5279 \IfStrEq{\LWR@strresult}{B}{B}{}%
5280 \IfStrEq{\LWR@strresult}{S}{r}{}%
5281 \IfStrEq{\LWR@strresult}{X}{p}{}%
5282 \LWR@traceinfo{lwarp@printmccoltype done}%
5283 }

```

`\LWR@multicolpartext` Print the data with paragraph tags:

```

5284 \newcommand*{\LWR@multicolpartext}{%
5285 \LWR@startpars%
5286 \LWR@multicoltext%
5287 \LWR@stoppars%
5288 }

```

`\LWR@multicolother` `{<colspec>}` For @, !, >, <, print the next token without paragraph tags:

```

5289 \newcommand*{\LWR@multicolother}[1]{%
5290 \addtocounter{LWR@tablemulticolspos}{1}%
5291 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
5292 \LWR@strresult%

```

A valid column data type was found:

```

5293 \booltrue{LWR@validtablecol}%
5294 }

```

`\LWR@multicolskip` Nothing to print for this column type.

```

5295 \newcommand*{\LWR@multicolskip}{%

```

A valid column data type was found:

```

5296 \booltrue{LWR@validtablecol}%
5297 }

```

`\LWR@printmccoldata` `{<colspec>}` Print the data for any valid column type found.

```
5298 \newcommand*{\LWR@printmccoldata}[1]{%
5299 \LWR@traceinfo{lw@printmccoldata -#1}%
```

Not yet found a valid column type:

```
5300 \boolfalse{\LWR@validtablecol}%
```

Get one token of the column spec:

```
5301 \StrChar{#1}{\the\LWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
5302 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
5303 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
5304 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
5305 \IfStrEq{\LWR@strresult}{D}{}%
5306 \addtocounter{\LWR@tablemulticolspos}{3}% skip parameters
5307 \LWR@multicoltext%
5308 }{}%
5309 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
5310 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
5311 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
5312 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
5313 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
5314 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
5315 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
5316 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
5317 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
5318 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{%
5319 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{%
5320 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{%
5321 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{%
```

If an invalid column type:

```
5322 \ifbool{\LWR@validtablecol}{\LWR@multicoltext}%
```

Tracing:

```
5323 \LWR@traceinfo{lw@printmccoldata done}%
5324 }
```

`\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`.

```
5325 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
5326 \setcounter{LWR@tablemulticolspos}{1}%
5327 \StrLen{#1}[\LWR@strresult]%
5328 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
5329 \whiledo{%
5330 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
5331 }%
5332 {%
```

Execute the assigned print function for each token in the column spec:

```
5333 #2{#1}%
```

Move to the next token in the column spec:

```
5334 \addtocounter{LWR@tablemulticolspos}{1}%
5335 }%
5336 }
```

58.17.2 Multicolumn factored code

```
5337 \newcommand{\LWR@multicoltext}{}
```

```
\LWR@domulticolumn [<1: vpos>] [<2: #rows>] [<3: numcols>] [<4: colspec>] [<5: text>]
```

```
5338 \NewDocumentCommand{\LWR@domulticolumn}{o o m m +m}{%
5339 \LWR@traceinfo{lw@domulticolumn -#1- -#2- -#3- -#4-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
5340 \renewcommand{\LWR@multicoltext}{%
5341 #5%
5342 \booltrue{LWR@validtablecol}%
5343 }%
```

Row processing:

```
5344 \LWR@maybenewtablerow%
5345 \LWR@tabularleftedge%
```

Begin the opening table data tag:

```
5346 \LWR@htmltag{td colspan="#3" %
5347 \IfValueT{#2}{ % rows?
5348 rowspan="#2" %

5349 \IfValueT{#1}{ % vpos?
5350 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{ }%
5351 \ifstrequal{#1}{t}{style="vertical-align:top" }{ }%
5352 }% vpos?
5353 }% rows?

5354 class="td%
```

Print the column type:

```
5355 \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”. Also add vertical bar class.

```
5356 \LWR@addcmidruletrim%
5357 \LWR@addleftmostbartag%
5358 \LWR@printbartag{\theLWR@tablecolspos}%
```

Close the class tag’s opening quote:

```
5359 "%
5360 \LWR@addcmidrulewidth%
5361 }% end of the opening table data tag
5362 \global\boolfalse{LWR@intabularmetadata}%
5363 \LWR@parsemulticolumnalignment{#4}{\LWR@printmccoldata}%
5364 }
```

58.17.3 Multicolumn

`\LWR@htmlmulticolumn` {<numcols>} {<alignment>} {<text>}

```
5365 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
5366 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5367 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn tag:

```
5368 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{#2}{#3}%
```

Move to the next \LaTeX column:

```
5369 \addtocounter{LWR@tablecolspos}{#1}%
5370 \addtocounter{LWR@tablecolspos}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
5371 \booltrue{LWR@skipatbang}%
5372 }
```

58.17.4 Longtable captions

longtable captions use \multicolumn.

Bool LWR@starredlongtable Per the caption package, step the counter if longtable*.

```
5373 \newbool{LWR@starredlongtable}
5374 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5375 \providecommand*\LTcapttype{table}
```

\LWR@longtabledatacaptiontag * [*<toc entry>*] {<caption>}

```
5376 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5377 {%
```

Remember the latest name for \nameref:

```
5378 \IfValueTF{#2}{% optional given?
5379   \ifthenelse{\equal{#2}{}}{% optional empty?
5380     {\LWR@setlatestname{#3}}% empty
5381     {\LWR@setlatestname{#2}}% given and non-empty
5382 }% optional given
5383 {\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:


```
5384 \LWR@tabularhtmlcolumns{1}{\theLWR@tabletotalcols}
```

Create the multicolumn tag:

```
5385 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{P}{% \LWR@domulticolumn
5386 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
5387 {% yes star
5388   \LWR@htmlblocktag{figcaption}%
5389   #3%
5390   \LWR@htmlblocktag{/figcaption}%
5391 }%
5392 {% No star:
```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```
5393   \ifbool{LWR@starredlongtable}%
5394   {%
5395   \ifthenelse{\equal{#2}{}}{% TOC entry
5396   }{%
5397   {%
5398     \refstepcounter{LTcaption}%
5399     \protected@edef\@currentlabel{%
5400       \csuse{p@LTcaption}\csuse{theLTcaption}}%
5401   }%
5402   }{%}
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
5403   \LWR@htmlblocktag{figcaption}%
5404   \csuse{fnum@LTcaption}\CaptionSeparator#3%
5405   \LWR@htmlblocktag{/figcaption}%
```

See if an optional caption was given:

```
5406   \ifthenelse{\equal{#2}{}}{% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
5407   }%
```

If the optional caption was given, but might only be []:

```
5408   {% TOC entry not empty
5409     \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```

5410      {% No TOC entry
5411          \addcontentsline%
5412          {\csuse{ext@\LTcapttype}}}%
5413          {\LTcapttype}%
5414          {%
5415          \protect\numberline%
5416          {\csuse{p@\LTcapttype}\csuse{the\LTcapttype}}}%
5417          {\ignorespaces #3\protect\relax}%
5418          }%
5419      }% end of No TOC entry

```

The optional caption has text enclosed:

```

5420      {% yes TOC entry
5421          \addcontentsline%
5422          {\csuse{ext@\LTcapttype}}}%
5423          {\LTcapttype}%
5424          {%
5425          \protect\numberline%
5426          {\csuse{p@\LTcapttype}\csuse{the\LTcapttype}}}%
5427          {\ignorespaces #2\protect\relax}%
5428          }%
5429      }% end of yes TOC entry
5430  }% end of TOC entry not empty
5431 }% end of no star

```

Skip any trailing @ or ! columns for this cell:

```

5432 \booltrue{LWR@skipatbang}%
5433 }% end of \LWR@domulticolumn
5434
5435 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
5436 \addtocounter{LWR@tablecolspos}{-1}
5437
5438 }

```

58.17.5 Counting HTML tabular columns

The \TeX 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 \TeX \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.

```
5439 \newcounter{LWR@tabhtmlcolindex}
5440 \newcounter{LWR@tabhtmlcolend}
5441 \newcounter{LWR@tabhtmlcoltotal}
```

`\LWR@tabularhtmlcolumns` $\{ \langle \textit{starting L\TeX column} \rangle \} \{ \langle \textit{number L\TeX columns} \rangle \}$

Compute the total number of HTML columns being spanned, considering the starting L^AT_EX table column and the number of L^AT_EX 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.

```
5442 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the L^AT_EX span, to which additional @ and ! columns may be added:

```
5443 \setcounter{LWR@tabhtmlcolindex}{#1}%
5444 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5445 \setcounter{LWR@tabhtmlcolend}{#1}%
5446 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

Walk across the L^AT_EX columns looking for @ and ! columns:

```
5447 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
```

Temporarily define a macro equal to the @ specification for this column:

```
5448 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}%
5449 \theLWR@tabhtmlcolindex}}%
```

If the @ specification is not empty, add to the count:

```
5450 \ifdefempty{\LWR@atbangspec}%
5451 {}%
5452 {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
5453 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}%
5454 \theLWR@tabhtmlcolindex}}%
5455 \ifdefempty{\LWR@atbangspec}%
5456 {}%
5457 {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Move to the next \LaTeX column:

```
5458 \addtocounter{LWR@tabhtmlcolindex}{1}%
5459 }% whiledo
5460 }
```

```
5461 \end{warpHTML}
```

58.18 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 174 for the HTML versions.

for HTML & PRINT: 5462 `\begin{warpall}`

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
                {<8:text>}
```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:

<https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex>

After the user may have

```
5463 \AtBeginDocument{
```

`\@ifundefined{@xmultirow}` determines if `multirow` was never loaded.

```
5464 \@ifundefined{@xmultirow}
5465 {}% no version of multirow was loaded
5466 {% \@xmultirow defined, so some version of multirow was loaded
```

`\@ifpackageloaded{multirow}` determines if v2.0 or later of `multirow` was used, which included the `\ProvidesPackage` macro.

```
5467 \@ifpackageloaded{multirow}{% v2.0 or newer
5468 \@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
5469 {% v2.0+:
5470 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5471 {+m +m +0{c} +m +0{0} +m +0{0pt} +m}%
5472 {\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}[#5]{#6}[#7]{#8}}}%
5473 }
5474 {% loaded but older, probably not executed:
```

```

5475 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5476     {+m +m +O{c} +m +O{0} +m +O{Opt} +m}%
5477 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
5478 }
5479 }% packageloaded{multirow}

```

If not `\@ifpackageloaded{multirow}` but `\@xmultirow` is defined, then this must be v1.6 or earlier, which did not `\ProvidesPackage{multirow}`, and did not have the `vposn` option.

```

5480 {% v1.6 or older did not \ProvidePackage
5481 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5482     {+m +m +O{c} +m +O{0} +m +O{Opt} +m}%
5483 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
5484 }

```

The user-level interface. This is provided if the HTML version was not already given.

```

5485 \providecommand*\multicolumnrow{\LWR@origmulticolumnrow}
5486 }% \@xmultirow defined, so multirow was loaded
5487 }% AtBeginDocument

5488 \end{warpall}

```

58.19 Utility macros inside a table

for HTML output: 5489 \begin{warpHTML}

```

5490 \newcommand*\LWR@donothing{}

```

In case bigdelim is not loaded:

```

5491 \newcommand*\ldelim{}
5492 \newcommand*\rdelim{}

5493 \end{warpHTML}

```

58.20 Special-case tabular markers


for HTML & PRINT: 5494 \begin{warpall}

`\TabularMacro` Place this just before inserting a custom macro in a table data cell. Doing so tells lwarp not to automatically start a new HTML table data cell yet. See section 7.7.

```
5495 \let\TabularMacro\relax
```

```
5496 \end{warpall}
```

`\ResumeTabular` Used to resume tabular entries after resuming an environment.

 **tabular inside another environment** When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing 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: 5497 `\begin{warpHTML}`

```
5498 \newcommand*{\ResumeTabular}{%
5499 \global\boolfalse{LWR@exitingtabular}%
5500 \global\boolfalse{LWR@tabularmutemods}%
5501 \LWR@getmynexttoken%
5502 }
```

```
5503 \end{warpHTML}
```

for PRINT output: 5504 `\begin{warpprint}`

```
5505 \newcommand*{\ResumeTabular}{{}
```

```
5506 \end{warpprint}}
```

58.21 Checking for a new table cell

for HTML output: 5507 \begin{warpHTML}

Bool LWR@exitingtabular When \end is found, turns off the next opening data tag.

5508 \newbool{LWR@exitingtabular}

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

5509 \newbool{LWR@tabularmutemods}

\LWR@tabledatacolumnntag Open a new HTML table cell unless the next token is for a macro which does not create data, such as \hline, \toprule, etc:

5510 \newcommand*{\LWR@tabledatacolumnntag}%

5511 {%

5512 \LWR@traceinfo{LWR@tabledatacolumnntag}%

\show\LWR@mynexttoken to see what tokens to look for

If not any of the below, start a new table cell:

5513 \let\mynext\LWR@tabledatasinglecolumnntag%

If exiting the tabular:

5514 \ifdefequal{\LWR@mynexttoken}{\end}%

5515 {\booltrue{LWR@exitingtabular}}{}%

longtable can have a caption in a cell

5516 \ifdefequal{\LWR@mynexttoken}{\caption}%

5517 {\let\mynext\LWR@donothing}{}%

Look for other things which would not start a table cell:

5518 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%

5519 {\let\mynext\LWR@donothing}{}%

5520 \ifdefequal{\LWR@mynexttoken}{\multirow}%

5521 {\let\mynext\LWR@donothing}{}%

5522 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%

5523 {\let\mynext\LWR@donothing}{}%

If an `\mrowcell`, this is a cell to be skipped over:

```
5524 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
5525     {\let\mynext\LWR@donothing}{}%
```

If an `\mcolrowcell`, this is a cell to be skipped over:

```
5526 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
5527     {\let\mynext\LWR@donothing}{}%
5528 %
5529 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
5530     {\let\mynext\LWR@donothing}{}%
5531 %
5532 \ifdefequal{\LWR@mynexttoken}{\hline}%
5533     {\let\mynext\LWR@donothing}{}%
5534 %
5535 \ifdefequal{\LWR@mynexttoken}{\toprule}%
5536     {\let\mynext\LWR@donothing}{}%
5537 %
5538 \ifdefequal{\LWR@mynexttoken}{\midrule}%
5539     {\let\mynext\LWR@donothing}{}%
5540 %
5541 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
5542     {\let\mynext\LWR@donothing}{}%
5543 %
5544 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
5545     {\let\mynext\LWR@donothing}{}%
5546 %
5547 \ifdefequal{\LWR@mynexttoken}{\cline}%
5548     {\let\mynext\LWR@donothing}{}%
5549 %
5550 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
5551     {\let\mynext\LWR@donothing}{}%
5552 %
5553 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
5554     {\let\mynext\LWR@donothing}{}%
5555 %
5556 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
5557     {\let\mynext\LWR@donothing}{}%
5558 %
5559 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
5560     {\let\mynext\LWR@donothing}{}%
5561 %
5562 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
5563     {\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:

```
5564 \mynext%
5565 }

5566 \end{warpHTML}
```

58.22 \mrowcell

for HTML & PRINT: 5567 \begin{warpall}


\mrowcell  The user must insert \mrowcell into any \multirow cells which must be skipped. This command has no action during print output.

```
5568 \newcommand*{\mrowcell}{}

5569 \end{warpall}
```

58.23 \mcolrowcell

for HTML & PRINT: 5570 \begin{warpall}

\mcolrowcell  The user must insert \mcolrowcell into any \multicolumnrow cells which must be skipped. This command has no action during print output.

```
5571 \newcommand*{\mcolrowcell}{}

5572 \end{warpall}
```

58.24 New \tabular definition

for HTML output: 5573 \begin{warpHTML}

These are default defininitions in case booktabs is not loaded, and are not expected to used, but must exist as placeholders.

```

5574 \newcommand*{\LWR@origtoprule}[1]{\hline}
5575 \newcommand*{\LWR@origmidrule}[1]{\hline}
5576 \LetLtxMacro\LWR@origcmidrule\cline
5577 \newcommand*{\LWR@origbottomrule}[1]{\hline}
5578 \newcommand*{\LWR@origaddlinespace}[1][1]{\hline}
5579 \newcommand*{\LWR@origmoreecmidrules}{}
5580 \newcommand*{\LWR@origspecialrule}[3]{\hline}

```

`\LWR@hline` The definition of `\hline` depends on whether `tabls` has been loaded. If so, optional space below the line may be specified, but will be ignored.

```

5581 \AtBeginDocument{
5582 \ifpackageloaded{lwarp-tabls}
5583 {\newcommand*{\LWR@hline}[1][1]{%
5584   \booltrue{\LWR@doinghline}%
5585   \LWR@getmynexttoken}}
5586 {\newcommand*{\LWR@hline}{%
5587   \booltrue{\LWR@doinghline}%
5588   \LWR@getmynexttoken}}
5589 }% AtBeginDocument

```

`\LWR@nullifyNoAutoSpacing` For `babel-french`, turn off auto spacing at the start of the tabular, then nullify the autospacing commands inside the tabular, since they were not compatible with the tabular column parsing code, which uses `xstring`.

```

5590 \AtBeginDocument{
5591 \ifundefined{frenchbsetup}%
5592 {% no babel-french
5593   \newcommand*{\LWR@nullifyNoAutoSpacing}{}
5594 }% no babel-french
5595 {% yes babel-french
5596   \newcommand*{\LWR@nullifyNoAutoSpacing}{%
5597     \NoAutoSpacing%
5598     \renewcommand*{\NoAutoSpacing}{}%
5599     \renewcommand*{\LWR@FBcancel}{}%
5600   }
5601 }% yes babel-french
5602 }% AtBeginDocument

```

Env `LWR@tabular` [*$\langle vertposition \rangle$*] [*$\langle colspecs \rangle$*]

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```

5603 \StartDefiningTabulars
5604
5605 \newenvironment*{\LWR@tabular}[2][1]

```

```

5606 {%
5607 \LWR@traceinfo{LWR@tabular started}%
5608 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```

5609 \global\boolfalse{LWR@startedrow}%

```

Not yet doing any rules:

```

5610 \global\boolfalse{LWR@doinghline}%
5611 \global\boolfalse{LWR@doingtbrule}%
5612 \global\boolfalse{LWR@doingcmidrule}%

```

For babel-french, turn off auto spacing one time, then nullify the autospacing commands since were not compatible with the tabular parsing code.

```

5613 \LWR@nullifyNoAutoSpacing%

```

Have not yet found the end of tabular command. Unmute the @ and ! columns.

```

5614 \boolfalse{LWR@exitingtabular}%
5615 \boolfalse{LWR@tabularmutemods}%

```

Create the table tag:

```

5616 \global\booltrue{LWR@intabularmetadata}%
5617 \LWR@forcenewpage
5618 \LWR@htmlblocktag{table}%

```

Parse the table columns:

```

5619 \LWR@parsetablecols{#2}%

```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```

5620 \LWR@stoppars%

```

Track column # for setting text-align:

```

5621 \setcounter{LWR@tablecolspos}{1}%

```

Start looking for midrules:

```

5622 \LWR@clearmidrules%

```

`\\` becomes a macro to end the table row:

```
5623 \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:

The optional parameter for `\hline` supports the `tabls` package.

```
5624 \LWR@traceinfo{\LWR@tabular: redefining macros}%
5625 \renewcommand*{\hline}{\LWR@hline}%
5626 %
5627 \RenewDocumentCommand{\cline}{m}%
5628 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%

5629 \DeclareDocumentCommand{\toprule}{o d()}
5630   {%
5631     \IfValueTF{##1}%
5632       {\LWR@docmidrule[##1](){1-\theLWR@tabletotalcols}}%
5633       {\booltrue{\LWR@doingtbrule}}}%
5634   \LWR@getmynexttoken}%
5635 %
5636 \DeclareDocumentCommand{\midrule}{o d()}%
5637   {%
5638     \IfValueTF{##1}%
5639       {\LWR@docmidrule[##1](){1-\theLWR@tabletotalcols}}%
5640       {\booltrue{\LWR@doinghline}}}%
5641   \LWR@getmynexttoken}%
5642 %
5643 \DeclareDocumentCommand{\cmidrule}{O{\LWR@cmidrulewidth} d() m}%
5644 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
5645 %
5646 \DeclareDocumentCommand{\bottomrule}{o d()}
5647   {%
5648     \IfValueTF{##1}%
5649       {\LWR@docmidrule[##1](){1-\theLWR@tabletotalcols}}%
5650       {\booltrue{\LWR@doingtbrule}}}%
5651   \LWR@getmynexttoken}%
5652 %
5653 \DeclareDocumentCommand{\addlinespace}{o}{}%
5654 \DeclareDocumentCommand{\morecmidrules}{-}{}%
5655 \DeclareDocumentCommand{\specialrule}{m m m d()}%
5656   {\LWR@docmidrule[##1](){1-\theLWR@tabletotalcols}\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`.

```

5657 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
5658 \renewcommand*{\mrowcell}{%
5659     \LWR@maybenewtablerow%
5660     \LWR@tabularleftedge%
5661     \global\booltrue{\LWR@skippingmrowcell}%
5662 }%
5663 \renewcommand*{\mcolrowcell}{%
5664     \LWR@maybenewtablerow%
5665     \LWR@tabularleftedge%
5666     \global\booltrue{\LWR@skippingmcolrowcell}%
5667 }%
5668 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%

```

Reset for new processing:

```

5669 \global\boolfalse{\LWR@tableparcell}%
5670 \global\boolfalse{\LWR@skippingmrowcell}%
5671 \global\boolfalse{\LWR@skippingmcolrowcell}%
5672 \global\boolfalse{\LWR@skipatbang}%
5673 \global\boolfalse{\LWR@emptyatbang}%

```

Set & for its special meaning inside the tabular:

```

5674 \StartDefiningTabulars%
5675 \protected\gdef&{\LWR@tabularampersand}%

```

Look ahead for a possible table data cell:

```

5676 \LWR@traceinfo{\LWR@tabular: about to \LWR@getmynexttoken}%
5677 \LWR@getmynexttoken%
5678 }%

```

Ending the environment:

```

5679 {%
5680 \LWR@traceinfo{\LWR@tabular ending}%
5681 \ifthenelse{\value{\LWR@tablecolspos}<\value{\LWR@tablecolswidth}}{%
5682     \LWR@tabularfinishrow%
5683 }%
5684 {%
5685     \LWR@closetabledatacell%
5686 }%
5687 \LWR@htmlblocktag{/tr}%
5688 \LWR@htmlblocktag{/table}%
5689 \global\boolfalse{\LWR@intabularmetadata}%

```

Unnest one level of tabular:

```

5690 \addtocounter{\LWR@tabulardepth}{-1}%

```

Restore & to its usual meaning:

```

5691 \protected\gdef&{\LWR@origampmacro}%
5692 \EndDefiningTabulars%
5693 \LWR@traceinfo{\LWR@tabular finished ending}%
5694 }
5695
5696 \EndDefiningTabulars

5697 \end{warpHTML}

```

58.25 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: 5698 \begin{warpHTML}

59.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

```
5699 \newcommand*{\@currentlabelname}{}

```

\LWR@stripperperiod {\langle text \rangle} [⟨.⟩]

Removes a trailing period.

```
5700 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%

```

Table 8: Cross-referencing data structures

Original \LaTeX:	(print and HTML)
<code>\refstepcounter:</code> Steps the couunter and sets <code>\@currentlabel</code> . <code>\@currentlabel:</code> <code>\p@<ctr>\the<ctr></code> Updated by <code>\refstepcounter</code> . <code>\label:</code> Writes to the .aux file: <code>\newlabel{<label>}{\@currentlabel}{\thepage}}</code> <code>\newlabel:</code> When the .aux file is read, sets <code>\r@<label></code> . <code>\r@<label>:</code> Set to: <code>{\@currentlabel}{\thepage}}</code> <code>\ref:</code> Returns the first part of <code>\r@<label></code> . <code>\pageref:</code> Returns the second part of <code>\r@<label></code> .	
Added by lwarp:	(HTML only)
<code>\label:</code> Adds HTML tags (section 59.3), plus <code>\splabel</code> data (section 59.2): <code>zLWR@name:</code> The section name for this label. <code>zLWR@htmlfilenumber:</code> The filenumber or name for this label. <code>zLWR@lateximagedepth:</code> The lateximagedepth for this label. <code>zLWR@lateximagenumber:</code> The lateximagenumber for this label. <code>\nameref:</code> Emualted from hyperref for lwarp. See section 59.4. <code>\ref</code> and <code>\nameref:</code> Adds HTML tags. See section 59.4.	
Added by amsmath:	(print and HTML)
<code>\label:</code> Execution is delayed until the math environment is completed. <code>\ltx@label:</code> \LaTeX <code>\label</code> , (HTML: patched by lwarp,) later patched by cleveref.	
Added by cleveref:	(print and HTML)
<code>\refstepcounter:</code> Added: sets <code>\cref@currentlabel</code> . <code>\cref@currentlabel:</code> (<code><type>=<ctr></code> unless an alias is used): <code>[<type>][\arabic{<ctr>}][<parent ctrs>]{\p@<ctr>\the<ctr>}</code> Also see section 45.4 for use with footnotes. <code>\label:</code> Writes to the .aux file: <code>\newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}</code> <code>\newlabel:</code> (Unchanged.) When the .aux file is read, sets <code>\r@<label>@cref</code> . <code>\r@<label>@cref:</code> Set to: <code>{\cref@currentlabel}{\thepage}}</code> Utility functions: See <code>\cref@getlabel</code> , <code>\cref@gettype</code> , <code>\cref@getcounter</code> , <code>\cref@getprefix</code> . Cross-referencing names: <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.	
Additionally patched by lwarp:	(HTML only)
<code>\cref, etc.:</code> Modified for lwarp. See section 69. <code>\label inside math:</code> See section 63.4.1.	
Footnotes: See <code>\noteentry</code> in section 45.4.	

`\LWR@setlatestname` $\{\langle object\ name\rangle\}$

Removes `\label`, strips any final period, and remembers the result.

```
5701 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `getttitlestring`.

```
5702 \GetTitleStringExpand{#1}%
5703 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5704 \edef\@currentlabelname{%
5705 \expandafter\LWR@stripperperiod\@currentlabelname%
5706 \ltx@empty.\ltx@empty\@nil%
5707 }%
5708 }
```

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`:

```
5709 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5710 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5711 \zref@newprop{zLWR@htmlfilenumber}{%
5712 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}}%
5713 }%
```

Additional properties for `lateximages`:

```
5714 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5715 \zref@newprop{zLWR@lateximagenumber}{\arabic{LWR@lateximagenumber}}
```

`zLWR@htmlfilenumber` property holds the file number or name

Add a `LWR@htmlfilenumber` property, and `lateximage` properties to `special`:


```

5716 \zref@addprop{special}{zLWR@name}
5717 \zref@addprop{special}{zLWR@htmlfilenumber}
5718 \zref@addprop{special}{zLWR@lateximagedepth}
5719 \zref@addprop{special}{zLWR@lateximagenumber}

```

Returns the selected field:

```

5720 \newcommand*{\LWR@sprof}[2]{%
5721 \zref@extractdefault{#1}{#2}{??}}

```

`\LWR@nameref` $\{\langle label \rangle\}$ Returns the section name for this label:

```

5722 \newcommand*{\LWR@nameref}[1]{%
5723 \LWR@sprof{#1}{zLWR@name}%
5724 }

```

`\LWR@htmlfileref` $\{\langle label \rangle\}$ Returns the file number for this label:

```

5725 \newcommand*{\LWR@htmlfileref}[1]{%
5726 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5727 \LWR@sprof{#1}{zLWR@htmlfilenumber}%
5728 }

```

`\LWR@lateximagedepthref` $\{\langle label \rangle\}$ Returns the lateximagedepth for this label:

```

5729 \newcommand*{\LWR@lateximagedepthref}[1]{%
5730 \LWR@sprof{#1}{zLWR@lateximagedepth}%
5731 }

```

`\LWR@lateximagenumberref` $\{\langle label \rangle\}$ Returns the lateximagenumber for this label:

```

5732 \newcommand*{\LWR@lateximagenumberref}[1]{%
5733 \LWR@sprof{#1}{zLWR@lateximagenumber}%
5734 }

```

`\LWR@splabel` $\{\langle label \rangle\}$ Sanitize the name and then creates the label:

```

5735 \newcommand*{\LWR@splabel}[1]{%
5736 \LWR@setlatestname{\@currentlabelname}%
5737 \zref@labelbylist{#1}{special}}

```

59.3 Labels

`\LWR@subsublabel` $\{\langle label \rangle\}$ Creates an HTML id tag.

```
5738 \newcommand*{\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5739 \ifthenelse{\cnttest{\value{\LWR@lateximagedepth}}{>}{0}}{%
5740 {}}%
5741 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
5742 \ifbool{\LWR@doingstartpars}%
5743 {% pars allowed
5744 \ifbool{\LWR@doingapar}
5745 {% par started
5746 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5747}% par started
5748 {% par not started
5749 \LWR@stoppars%
5750 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5751 \LWR@startpars%
5752}% par not started
5753}% pars allowed
5754 {% pars not allowed
5755 \LWR@htmltag{a id="#1"{}}\LWR@htmltag{/a}%
5756}% pars not allowed
5757}% not lateximage
5758 }
```

`\LWR@newlabel` $\{\langle label \rangle\}$ [$\langle type \rangle$]

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular \LaTeX `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional $\langle type \rangle$ is per the `ntheorem` package, and is ignored.

```
5759 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5760 \LWR@traceinfo{\LWR@newlabel: starting}%
5761 \LWR@traceinfo{\LWR@newlabel: !#1!}%
5762 % \@bsphack%
```

Create a traditional \LaTeX label, as modified by cleveref:

```
5763 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, `LWR@htmlfilename`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```
5764 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5765 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5766 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilename is \theLWR@htmlfilename}%
5767 \LWR@splabel{#1}%
5768 \LWR@subsublabel{#1}%
5769 % \@esphack%
5770 \LWR@traceinfo{LWR@newlabel: done}%
5771 }
```

59.4 References

`\LWR@startref` `{\label{}}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```
5772 \newcommand*{\LWR@startref}[1]
5773 {%
5774 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
5775 \LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
5776 \LWR@htmltag{a href="%
5777 \LWR@traceinfo{LWR@startref B}%
5778 \LWR@htmlrefsectionfilename{#1}%
5779 \LWR@traceinfo{LWR@startref C}%
5780 \#%
```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```
5781 \LWR@traceinfo{LWR@startref D: !#1!}%
5782 \ifthenelse{\equal{\LWR@lidref}{??}}%
```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

5783 {%
5784   \LWR@traceinfo{LWR@startref D0: ??}%
5785   ??%
5786 }%

```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```

5787 {%
5788   \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
5789   \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
5790   {%
5791     \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
5792     lateximage\LWR@lateximagenumberref{#1}%
5793   }%
5794   {%
5795     \LWR@traceinfo{LWR@startref D3}%
5796     #1%
5797   }%
5798 }%
5799 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

5800 "}}}%
5801 \LWR@traceinfo{LWR@startref F}%
5802 }

```

`\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)”.

```

5803 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5804 \LWR@traceinfo{LWR@subnewref #1 #2}%
5805 \LWR@startref{#1}%
5806 \LWR@origref{#2}%
5807 \LWR@htmltag{/a}%
5808 }

```

`\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.

```

5809 \NewDocumentCommand{\LWR@newref}{s m}{%
5810 \LWR@traceinfo{LWR@newref #2}%

```

```

5811 \IfBooleanTF{#1}%
5812 {\LWR@origref{#2}}%
5813 {\LWR@subnewref{#2}{#2}}%
5814 }

```

`\pagerefPageFor` Text for page references.

```

5815 \newcommand*{\pagerefPageFor}{see }

```

`\pageref` * `{\label}` Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```

5816 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5817 \IfBooleanTF{#1}%
5818 {(\pagerefPageFor\LWR@origref{#2})}%
5819 {(\cpageref{#2})}%
5820 }

```

`\nameref` `{\label}`

```

5821 \newcommand*{\nameref}[1]{%
5822 \LWR@traceinfo{nameref A}%
5823 \LWR@startref{#1}%
5824 \LWR@traceinfo{nameref B}%
5825 \LWR@nameref{#1}%
5826 \LWR@traceinfo{nameref C}%
5827 \LWR@htmltag{/a}%
5828 \LWR@traceinfo{nameref D}%
5829 }

```

`\Nameref` `{\label}` In print, adds the page number. In HTML, does not.

```

5830 \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.

```
5831 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
5832 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref` $\{\langle URL \rangle\} \{\langle text \rangle\}$

```
5833 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5834 \LWR@htmltag{a href="#1" target="_{}blank"\LWR@orignewline}#2\LWR@htmltag{/a}%
5835 \LWR@ensuredoingapar%
5836 }
```

`\LWR@subhyperrefclass` $\{\langle URL \rangle\} \{\langle text \rangle\} \{\langle htmlclass \rangle\}$

```
5837 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5838 \LWR@htmltag{a href="#1"
5839 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5840 \LWR@ensuredoingapar%
5841 }
```

`\href` $[\langle options \rangle] \{\langle URL \rangle\} \{\langle text \rangle\}$

Create a link with accompanying text:

```
5842 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
5843 \LWR@subhyperref{#2}{#3}%
5844 \endgroup%
5845 \LWR@ensuredoingapar%
5846 }
5847
5848 \newcommand{\href}{%
5849 \LWR@ensuredoingapar%
5850 \begingroup%
5851 \catcode'\_ =12
5852 \LWR@hrefb%
5853 }
```

`\nolinkurl` $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```

5854 \newcommand*{\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
5855
5856 \newcommand{\nolinkurl}{\%
5857 \LWR@ensuredoingapar%
5858 \begingroup\catcode'\_ =12
5859 \LWR@nolinkurlb%
5860 }

```

`\url` `{\langle URL \rangle}`

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`.

```

5861 \newcommand*{\LWR@urlb}[1]{\%
5862 \href{#1}{#1}%
5863 \endgroup%
5864 \LWR@ensuredoingapar%
5865 }
5866
5867 \newcommand{\LWR@urla}{\%
5868 \LWR@ensuredoingapar%
5869 \begingroup\catcode'\_ =12
5870 \LWR@urlb%
5871 }
5872
5873 \let\url\LWR@urla

```

`\LWR@subinlineimage` `[\langle alttag \rangle]{\langle class \rangle}{\langle filename \rangle}{\langle extension \rangle}{\langle style \rangle}`

```

5874 \newcommand*{\LWR@subinlineimage}[5][\%
5875 \ifthenelse{\equal{#1}{}}{\%
5876 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}{}}}%
5877 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}{}}}%
5878 }

5879 \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 remembering whether a `\listof` was requested for a float of this type.

File with extension `lo<f,t,a-z>`: An output file containing the commands to build the `\listof<type><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: 5880 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
5881 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5882 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5883
5884 === #1 begin
5885
5886 }{}%
5887 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5888 \addtocounter{LWR@thisfloat}{1}%
5889 \booltrue{LWR@freezethisfloat}%
```

```
5890 \begingroup
```

Settings while inside the environment:

```
5891 \LWR@origraggedright
```

Open an HTML figure tag:

```
5892 \LWR@htmltag{figure id="autofloat-\arabic{LWR@thisfloat}" class="#1"}
```

```
5893 \renewcommand*{\@capttype}{#1}
5894 \caption@settype{#1}
5895 \LWR@startpars
5896 }
```

\@float Support packages which create floats directly.
 \@dblfloat

```
5897 \let\@float\LWR@floatbegin
5898 \let\@dblfloat\LWR@floatbegin
```

\LWR@floatend Ends a \newfloat environment.

```
5899 \newcommand*{\LWR@floatend}{%
5900 \LWR@stoppars%
5901 \LWR@htmllementend{figure}%
```

```

5902 \endgroup%
5903 \boolfalse{LWR@freezethisfloat}%
5904 \LWR@startpars%
5905 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5906
5907 === end
5908
5909 }{}%
5910 }

```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```

5911 \let\end@float\LWR@floatend
5912 \let\end@dblfloat\LWR@floatend

```

Ctrl `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.

```

5913 \newcounter{LWR@thisfloat}

```

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```

5914 \newbool{LWR@freezethisfloat}
5915 \boolfalse{LWR@freezethisfloat}

```

`\LWR@maybeinthisfloat`

```

5916 \newcommand*{\LWR@maybeinthisfloat}{%
5917 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
5918 }

```

`\@capttype` Remembers which float type is in use.

```

5919 \newcommand*{\@capttype}{}

```

60.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

5920 \AtBeginDocument{\providecommand*\CaptionSeparator{:-}}

```

`\@makecaption` `{<name and num>}{<text>}`

Prints the float type and number, the caption separator, and the caption text.

```

5921 \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.

Ctrl `LWR@nextautofloat` Tracks `autofloat` for floats. Tracks `autopage` for floats.

Ctrl `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
5922 \newcounter{LWR@nextautofloat}
5923 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat` `{\langle autopage \rangle}{\langle autofloat \rangle}`

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.

```
5924 \newcommand*{\LWRsetnextfloat}[2]{%
5925 \setcounter{LWR@nextautopage}{#1}%
5926 \setcounter{LWR@nextautofloat}{#2}%
5927 }
```

Ctrl `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.

```
5928 \newcounter{LWR@latestautopage}
5929 \setcounter{LWR@latestautopage}{1}

5930 \let\LWR@origcaption@begin\caption@begin
5931 \let\LWR@origcaption@end\caption@end
5932 \let\LWR@orig@@par\@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
5933 \newcommand{\LWR@caption@begin}
5934 {
5935 \LWR@traceinfo{LWR@caption@begin}%
```

Keep `par` and `minipage` changes local:

```
5936 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5937 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
5938 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
5939 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
5940 \LWR@htmlblocktag{figcaption}%
5941 \LWR@origcaption@begin%
5942 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5943 \newcommand{\LWR@caption@end}
5944 {%
5945 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5946 \LWR@origvspace*{\baselineskip}%
```

Closing tag:

```
5947 \LWR@htmlblocktag{/figcaption}%
5948 \endgroup%
5949 % \leavevmode% avoid bad space factor (0) error
5950 \LWR@traceinfo{\LWR@caption@end: done}%
5951 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.
`\caption@end`

```
5952 \AtBeginDocument{
5953 \let\caption@begin\LWR@caption@begin
5954 \let\caption@end\LWR@caption@end
5955 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
5956 \let\LWR@origcaptionlistentry\captionlistentry
5957
5958 \renewcommand*{\captionlistentry}{%
```

```

5959 \LWR@maybeinthisfloat%
5960 \LWR@ensuredoingapar%
5961 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5962 \LWR@origcaptionlistentry%
5963 }
5964
5965 \def\LWR@LTcaptionlistentry{%
5966 \LWR@ensuredoingapar%
5967 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5968 \bgroup
5969 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
5970 {\egroup\LWR@LTcaptionlistentry}}%
5971 \def\LWR@LTcaptionlistentry#1{%
5972 \caption@listentry\@firstoftwo[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.

```

5973 \let\LWR@origaddcontentsline\addcontentsline
5974
5975 \renewcommand*{\addcontentsline}[3]{%
5976 \ifthenelse{\equal{#1}{toc}}{ }{%
5977 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
5978 \addtocontents{\@nameuse{ext@#2}}{%
5979 \protect\LWRsetnextfloat%
5980 {\arabic{LWR@latestautopage}}%
5981 {\arabic{LWR@thisfloat}}%
5982 }% addtocontents
5983 }% not toc
5984 \LWR@origaddcontentsline{#1}{#2}{#3}%
5985 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5986 \AtBeginDocument{
5987 \let\LWR@origcaptionof\captionof
5988
5989 \renewcommand*{\captionof}{%
5990 \LWR@maybeinthisfloat%
5991 \LWR@stoppars
5992 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5993 \LWR@origcaptionof%
5994 }
5995 }

5996 \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 \TeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 5997 `\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 \TeX , 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.

```
5998 \newcommand*{\LWR@myshorttoc}[1]{
5999 \LWR@ensuredoingapar
```

Only if the file exists:

```
6000 \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`.

```
6001 \begingroup
6002 % \ifxetexorluatex%
6003 % \else
6004 % \inputencoding{latin1}% currently disabled
6005 % \fi
```

6006 \makeatletter

Read in the toc file:

```
6007 \@input{\jobname.#1}
6008 % \makeatother
6009 \endgroup
6010 }%
6011 {}%
6012 }
```

\LWR@subtableofcontents {<toc/lof/lot>} {<sectionstarname>}

Places a TOC/LOF/LOT at the current position.

6013 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%

Closes previous levels:

```
6014 \@ifundefined{chapter}
6015 {\LWR@closeprevious{\LWR@depthsection}}
6016 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

6017 \LWR@printpendingfootnotes

Place the list into its own chapter (if defined) or section:

6018 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}

Create a new HTML nav containing the TOC/LOF/LOT:

6019 \LWR@htmlclass{nav}{#1}

Create the actual list:

6020 \LWR@myshorttoc{#1}

Close the nav:

```
6021 \LWR@htmlclassend{nav}{#1}
6022 }
```

Patch \@starttoc to encapsulate the TOC inside HTML tags:

6023 \let\LWR@orig@starttoc\LWR@starttoc

```

6024
6025 \renewcommand{\@starttoc}[1]{
6026 \LWR@htmlclass{nav}{#1}
6027 \LWR@orig@starttoc{#1}
6028 \LWR@htmlclassend{nav}{#1}
6029 }

```

Patch \tableofcontents, etc. to print footnotes first. newfloat uses \listoffigures for all future float types.

```

6030 \let\LWR@origtableofcontents\tableofcontents
6031 \let\LWR@origlistoffigures\listoffigures
6032 \let\LWR@origlistoftables\listoftables
6033
6034 \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:

```

6035 \ifbool{FormatWordProcessor}{}{

```

Copy the .toc file to .sidetoc for printing the sideroc. The original .toc file is renewed when \tableofcontents is finished.

```

6036 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
6037 \LWR@printpendingfootnotes
6038 \LWR@origtableofcontents
6039 }
6040 }
6041 \renewcommand*{\listoffigures}{
6042 \ifbool{FormatWordProcessor}{}{
6043 \LWR@printpendingfootnotes
6044 \LWR@origlistoffigures
6045 }
6046 }
6047
6048 \renewcommand*{\listoftables}{
6049 \ifbool{FormatWordProcessor}{}{
6050 \LWR@printpendingfootnotes
6051 \LWR@origlistoftables
6052 }
6053 }

```

61.2 High-level TOC commands

`\listof {<type>} {<title>}`

Emulate the `\listof` command from the float package (section 128). Used to create lists of custom float types. Also used to redefine the standard \TeX `\listoffigures` and `\listoftables` commands.

```
6054 \NewDocumentCommand{\listof}{m +m}{%
6055 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
6056 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
6057 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
6058     \jobname.\csuse{ext@#1}\relax
6059 }
```

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 sideroc.

div.sidetoctitle: The title.

div.sidetoccontents: The table of contents.

```
6060 \end{warpHTML}
```

for HTML & PRINT: 6061 \begin{warpall}

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard \TeX section level similar to `tocdepth`.

```
6062 \newcounter{SideTOCDepth}
6063 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideroc.

```
6064 \newcommand{\sidetocname}{Contents}
```

```
6065 \end{warpall}
```

for HTML output: 6066 \begin{warpHTML}

`\LWR@sidetoc` Creates the actual side-TOC.

```
6067 \newcommand*{\LWR@sidetoc}{
6068 \LWR@forcenewpage
6069 \LWR@stoppars
6070 }
```

The entire sidetoc is placed into a nav of class sidetoc.

```
6071 \LWR@htmlelementclass{nav}{sidetoc}
6072
6073 \setcounter{tocdepth}{\value{SideTOCDepth}}
6074 }
```

The title is placed into a <div> of class sidetoc title, and may contain paragraphs.

```
6075 \begin{BlockClass}{sidetoc title}
6076 \sidetocname
6077 \end{BlockClass}
```

The table of contents is placed into a <div> of class sidetoc contents.

```
6078 \begin{BlockClass}{sidetoc contents}
6079 \LinkHome
6080
6081 \LWR@myshorttoc{sidetoc}
6082 \end{BlockClass}
6083 \LWR@htmlelementclassend{nav}{sidetoc}
6084 }
```

61.4 Low-level TOC line formatting

`\numberline` $\{\langle number \rangle\}$

(Called from each line in the .aux, .lof files.)

Record this section number for further use:

```
6085 \renewcommand*{\numberline}[1]{%
6086 \LWR@sectionnumber{#1}\quad%
6087 }
```

`\hypertoc` $\{\langle 1: depth \rangle\} \{\langle 2: type \rangle\} \{\langle 3: name \rangle\} \{\langle 4: page \rangle\}$

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

```
6088 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
6089 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}}{%
6090   \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
6091   \LWR@subhyperrefclass{%
6092   \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
6093   \LWR@stoppars%
6094 }
6095 {}
6096 }
```

Ctrl lofdepth TOC depth for figures.

```
6097 \newcounter{lofdepth}
6098 \setcounter{lofdepth}{1}
```

Ctrl lotdepth TOC depth for tables.

```
6099 \newcounter{lotdepth}
6100 \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

```
6101 \newcommand{\hypertocfloat}[5]{%
6102 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
6103 \@ifundefined{c@#3depth}{%
6104 \newcounter{#3depth}%
6105 \setcounter{#3depth}{1}%
6106 }{}%
```

Respond to `lofdepth`, etc.:

```
6107 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
6108 \ifthenelse{\cnttest{#1}<=}{\arabic{#3depth}}{}%
6109 \LWR@startpars%
```

Create an HTML link to `filename#autofloat`-(float number), with text of the caption, of the given HTML class.

```
6110 \LWR@subhyperrefclass{%
6111 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
6112 \#autofloat-\arabic{LWR@nextautofloat}}%
6113 {#4}{toc#2}%
6114 \LWR@stoppars%
6115 }{}%
6116 }
```

Automatically called by `\contentsline`:

```
6117 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
6118 \DeclareDocumentCommand{\l@chapter}{m m}
6119 {\hypertoc{0}{chapter}{#1}{#2}}
6120 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
6121 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
6122 \renewcommand{\l@subsubsection}[2]
6123 {\hypertoc{3}{subsubsection}{#1}{#2}}
6124 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
6125 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
6126 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
6127 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

6128 \end{warpHTML}
```

62 Index and glossary

See:

<http://tex.stackexchange.com/questions/187038/>

[how-to-mention-section-number-in-index-created-by-imakeidx](#)

Index links are tracked by the counter `LWR@autoindex`. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output: 6129 `\begin{warpHTML}`

```
6130 \newcounter{LWR@autoindex}
6131 \setcounter{LWR@autoindex}{0}
6132
6133 \newcounter{LWR@autoglossary}
6134 \setcounter{LWR@autoglossary}{0}
```

`\printindex`

```
6135 \let\LWR@origprintindex\printindex
6136
6137 \renewcommand*\printindex{
6138 {
6139 \LWR@startpars
6140 \LWR@origprintindex
6141 }
```

Env `theindex`

```
6142 \@ifundefined{chapter}{
6143 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}}
6144 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}}
6145
6146 \renewenvironment*{theindex}{%
6147 \LWR@indexsection{\indexname}%
6148 \let\item\LWR@indexitem%
6149 \let\subitem\LWR@indexsubitem%
6150 \let\subsubitem\LWR@indexsubsubitem%
6151 }{}
```

`\LWR@indexitem`

```
6152 \newcommand*\LWR@indexitem{
6153
```

```
6154 \InlineClass{indexitem}{%}
6155 }
```

\LWR@indexitem

```
6156 \newcommand{\LWR@indexsubitem}{%
6157
6158 \InlineClass{indexsubitem}{%}
6159 }
```

\LWR@indexitem

```
6160 \newcommand{\LWR@indexsubsubitem}{%
6161
6162 \InlineClass{indexsubsubitem}{%}
6163 }
```

\@wrindex $\langle term \rangle$ Redefined to write the LWR@latestautopage counter instead of page

```
6164 \def\@wrindex#1{%
6165 \addtocounter{LWR@autoindex}{1}%
6166 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
6167 \protected@write\@indexfile{%
6168 {\string\indexentry{#1}{\theLWR@autoindex}}%
6169 \endgroup
6170 \@esphack}
```

\@wrglossary $\langle term \rangle$ Redefined to write the LWR@latestautopage counter instead of page

```
6171 \def\@wrglossary#1{%
6172 \addtocounter{LWR@autoglossary}{1}%
6173 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
6174 \protected@write\@glossaryfile{%
6175 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
6176 \endgroup
6177 \@esphack}
```

\hyperindexref $\langle autosecnumber \rangle$

\hyperindexref{web address} is inserted into *.ind by the xindy style file lwarp.xdy

```
6178 \newcommand*{\hyperindexref}[1]{\nameref{LWRindex-#1}}

6179 \end{warpHTML}
```


for PRINT output: A null command for print mode, in case hyperref was not used:

```
6180 \begin{warpprint}
6181 \newcommand{\hyperindexref}[1]{#1}
6182 \end{warpprint}
```

for HTML & PRINT: For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
6183 \begin{warppall}
6184 \providecommand{\glo@name}{}
6185 \end{warppall}
```

63 Math

 **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.

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 \LaTeX 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 \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the \LaTeX code which generated the math, allowing copy/paste of the \LaTeX math expression into other documents.

SVG image font size 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 240 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 \LaTeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of \LaTeX , and
2. As detokenized printed \LaTeX commands placed directly into the HTML output for interpretation by the MathJax display scripts. An additional script is used to pre-set the equation number format and value according to the current \LaTeX values, and the MathJax cross-referencing system is ignored in favor of the \LaTeX internal system, seamlessly integrating with the rest of the \LaTeX code.

MathJax limitations Limitations when using MathJax include:

Prog MathJax

chapter numbers

- In document classes which have chapters, $\backslash\text{tagged}$ equations have the chapter number prepended in HTML output, unlike \LaTeX . $\backslash\text{tag*}$ 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 $\backslash\text{SI}$ or $\backslash\text{si}$ inside $\backslash\text{text}$, where it will be rendered as normal text.

<https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx>

 **siunitx inside an equation**

L^AT_EX macros

- MathJax does not automatically support custom L^AT_EX macros, but they may be set up by the user.

custom MathJax macros

As an example of using custom L^AT_EX 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: 6186 `\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.

```
6187 \let\LWR@origtextdollar\$
6188
6189 \renewcommand*{\$}{%
6190 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}{%
6191 {\LWR@origtextdollar}%
6192 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
6193 }
```

`\LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
6194 \newcounter{LWR@externalfilecnt}
```

63.1 Inline and display math

```
6195 \LetLtxMacro\LWR@origdollar$
6196 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

6197 \LetLtxMacro\LWR@origopenparen\(  
6198 \LetLtxMacro\LWR@origcloseparen\)
```

```
6199 \LetLtxMacro\LWR@origopenbracket\[
6200 \LetLtxMacro\LWR@origclosebracket\]
```

\$ Redefine the dollar sign to place math inside a lateximage, or use MathJax:
\$\$

```
6201 \begingroup
6202 \catcode'\$=\active%
6203 \protected\gdef$\{\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

\LWR@doubledollar Redefine the double dollar sign to place math inside a lateximage, or use MathJax:

```
6204 \gdef\LWR@doubledollar$#1$${
6205 \ifbool{mathjax}%

```

For MathJax, print the math between $\[$ and $\]$:

```
6206 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}
```

For SVG, print the math inside a lateximage, with an ALT tag of the \LaTeX code:

```
6207 {% not mathjax
6208
6209 \begin{lateximage}%
6210 [\textbackslash{ } \LWR@HTMLsanitize{#1} \textbackslash{ } ]%
6211 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
6212 \end{lateximage}
6213
6214 }
6215 }%
```

\LWR@singledollar Redefine the single dollar sign to place math inside a lateximage, or use MathJax:

```
6216 \gdef\LWR@subsingledollar#1{%
6217 \ifbool{mathjax}%

```

For MathJax, print the math between $\($ and $\)$:

```
6218 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}
```

For SVG, print the math inside a lateximage, with an ALT tag of the \LaTeX code:

```
6219 {% not mathjax
6220 \begin{lateximage}%
6221 [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash )]%
6222 \LWR@origdollar#1\LWR@origdollar%
6223 \end{lateximage}%
6224 }%
6225 }
```

```

6226
6227 \gdef\LWR@singledollar#1${%
6228 \LWR@subsingledollar{#1}%
6229 }

```

\(Redefine to the above dollar macros.
 \)

```

6230 \gdef\(#1\){$#1$}
6231 \gdef\[#1\]{$$#1$$}
6232
6233 \endgroup
6234
6235
6236

```

\@ensuredmath $\langle expression \rangle$ Not yet adapted to lwarp.

```

6237 \let\LWR@orig@ensuredmath\@ensuredmath
6238
6239 \newcommand*{\LWR@ensuredmath}[1]{%
6240 \LWR@traceinfo{\@ensuredmath: page \thepage: \detokenize{#1}}%
6241 \LWR@subsingledollar{#1}%
6242 }

```

Remove the old math and displaymath environments:

```

6243 \let\math\relax
6244 \let\endmath\relax
6245 \let\displaymath\relax
6246 \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.

```

6247 \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.

```

6248 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}

```

63.2 MathJax support

Ctrl **LWR@nextequation** Used to add one to compute the next equation number.

```
6249 \newcounter{LWR@nextequation}
```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\(" and “\)” characters. They are printed to HTML output, not interpreted by \LaTeX .

```
6250 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
6251 \ifcsdef{thechapter}{
6252 \InlineClass{hidden}{
6253 \textbackslash(
6254 \textbackslash{seteqsection \{\thechapter\}
6255 \textbackslash)}
6256 }
6257 }
6258 {}}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
6259 \ifthenelse{\cnttest{\value{equation}}>0}{
6260 {
```

Tell MathJax that the next set of equations begins with the current \LaTeX equation number, plus one.

```
6261 \setcounter{LWR@nextequation}{\value{equation}}
6262 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\(" and “\)” characters, to be printed to HTML, not interpreted by \LaTeX .

```
6263 \InlineClass{hidden}{
6264 \textbackslash(
6265 \textbackslash{seteqnumber \{\arabic{LWR@nextequation}\}
6266 \textbackslash)}
6267 }
6268 }{}% not eq > 0
6269 }
```

`\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`.

```
6270 \newcommand*{\LWR@restoreorigformatting}{%
6271 \LWR@traceinfo{\LWR@restoreorigformatting}%
6272 \RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
6273 \RenewDocumentEnvironment{BlockClass}{o m}{}{}%
6274 \renewcommand{\BlockClassSingle}[2]{##2}%
6275 \LetLtxMacro{\hspace}{\LWR@origspace}%
6276 \LetLtxMacro\rule\LWR@origrule%
6277 \let\,\LWR@origcomma% disable HTML short unbreakable space
6278 \let\textellipsis\LWR@origtextellipsis%
6279 \let\textless\LWR@origtextless%
6280 \let\textgreater\LWR@origtextgreater%
6281 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
6282 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
6283 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
6284 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
6285 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
6286 \LetLtxMacro{\textit}{\LWR@origtextit}%
6287 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
6288 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
6289 \LetLtxMacro{\textup}{\LWR@origtextup}%
6290 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
6291 \LetLtxMacro{\emph}{\LWR@origemph}%
6292 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
6293 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
6294 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
6295 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
6296 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
6297 \LetLtxMacro{\upshape}{\LWR@origupshape}%
6298 \LetLtxMacro{\slshape}{\LWR@origslshape}%
6299 \LetLtxMacro{\scshape}{\LWR@origscshape}%
6300 \LetLtxMacro{\itshape}{\LWR@origitshape}%
6301 \LetLtxMacro{\em}{\LWR@origem}%
6302 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
6303 \let\sp\LWR@origsp%
6304 \let\sb\LWR@origsb%
6305 \let\textsuperscript\LWR@origtextsuperscript%
6306 \let\textsubscript\LWR@origtextsubscript%
6307 \LetLtxMacro\underline\LWR@origunderline%
6308 \let~\LWR@origtilde%
6309 \let\enskip\LWR@origenskip%
6310 \let\quad\LWR@origquad%
6311 \let\qquad\LWR@origqquad%
6312 \LetLtxMacro{\tabular}{\LWR@origtabular}%
6313 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
6314 \LetLtxMacro\toprule\LWR@origtoprule%
6315 \LetLtxMacro\midrule\LWR@origmidrule%
6316 \LetLtxMacro\cmidrule\LWR@origcmidrule%
6317 \LetLtxMacro\bottomrule\LWR@origbottomrule%
6318 \LetLtxMacro\addlinespace\LWR@origaddlinespace%
6319 \LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
```

```

6320 \LetLtxMacro\specialrule\LWR@origspecialrule%
6321 \let\newline\LWR@orignewline%
6322 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
6323 \LetLtxMacro\includegraphics\LWR@origincludegraphics%
6324 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
6325 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
6326 \let\reflectbox\LWR@origreflectbox%
6327 \LetLtxMacro\resizebox\LWR@origresizebox%
6328 \let\framebox\LWR@origframebox%
6329 \let\makebox\LWR@origmakebox%
6330 \let\fbbox\LWRprint@fbbox%
6331 \let\fbboxBlock\LWRprint@fbbox%
6332 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
6333 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
6334 \LetLtxMacro{\minipage}{\LWR@origminipage}%
6335 \let\endminipage\LWR@endminipage%
6336 \LetLtxMacro{\parbox}{\LWR@origparbox}%
6337 \let\TeX\LWR@origTeX%
6338 \let\LaTeX\LWR@origLaTeX%
6339 \let\LaTeXe\LWR@origLaTeXe%
6340 \renewcommand*{\Xe}{X\textsubscript{E}}%
6341 %
6342 \LWR@restoreorigaccents%
6343 %
6344 \LWR@FBcancel%
6345 }

```

`\LWR@hidelatexequation` $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Creates the \LaTeX version of the equation inside an HTML comment.

```
6346 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```

6347 \LWR@stoppars
6348 \LWR@htmlopencomment
6349

```

Start the \LaTeX math environment inside the HTML comment:

```

6350 \begingroup
6351 \csuse{\LWR@orig#1}

```

While in the math environment, restore various commands to their \LaTeX meanings.

```
6352 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section [63.4.1](#).

Print the contents of the equation:

6353 #2

End the \LaTeX math environment inside the HTML comment:

```
6354 \csuse{LWR@origend#1}
6355 \endgroup
6356
```

Close the HTML comment and resume HTML paragraph handling:

```
6357 \LWR@htmlclosecomment
6358 \LWR@startpars
6359 }
```

```
\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 .

```
6360 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed “\ (” and “\)” characters.

```
6361 \LWR@origtilde\LWR@orignewline
6362 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
6363 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
6364 \textbackslash{}end\{#1\}
6365 \LWR@orignewline
6366 }
```

63.3 Equation environment

Remember existing equation environment:

```
6367 \let\LWR@origequation\equation
6368 \let\LWR@origendequation\endequation
```

Remove existing equation environment:

```
6369 \let\equation\relax
6370 \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 \TeX 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.

```
6371 \NewEnviron{equation}{%
6372
6373 \ifbool{mathjax}
```

MathJax output:

```
6374 {
```

Print commands to synchronize MathJax's equation number and format to the current \TeX chapter/section and equation number:

```
6375     \LWR@syncmathjax
```

Print the \TeX math inside an HTML comment:

```
6376     \LWR@hidelatexequation{equation}{\BODY}
6377 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the \TeX equation environment commands, and the contents of the environment's `\BODY`.

```
6378 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
6379     \begin{lateximage}[ $\theequation$  \textbackslash{begin\{equation\}}}%
6380     \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}}%
6381     \textbackslash{end\{equation\}}]% alt tag
```


Create the actual \LaTeX -formatted equation inside the `lateximage` using the contents of the environment.

```
6382 \LWR@origequation
6383 \BODY% contents collected by NewEnviron
6384 \LWR@origendequation
6385 \end{lateximage}%
6386 }
6387
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
6388 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

63.4 AMS Math environments

63.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for multiline-specific code, `LWR@amsmultline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a multiline environment and not an equation.

```
6389 \newbool{LWR@amsmultline}
6390 \boolfalse{LWR@amsmultline}
```

`\LWR@htmlmathlabel` $\{\langle label \rangle\}$

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a \LaTeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the \LaTeX original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
6391 \newcommand*{\LWR@htmlmathlabel}[1]{%
6392 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
6393 \ifbool{mathjax}{%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
6394 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6395 \ifbool{LWR@amsmultline}{\hspace*{\totwidth@}}
```

Temporarily end the HTML comment, insert the \LaTeX & HTML label, then resume the HTML comment. `\@firstofone` is required to remove extra braces introduced by the `amsmath` package.)

```
6396 \LWR@htmlclosecomment%
6397 \LWR@origltx@label{#1}%
6398 \LWR@htmlopencomment%
6399 }% text
6400 }% mathjax
6401 {%
6402 \LWR@origltx@label{#1}%
6403 }%
6404 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
6405 \newcommand*{\LWR@beginhideamsmath}{
6406 \LWR@stoppars
6407 \LWR@origtilde\LWR@orignewline
6408 \LWR@htmlopencomment
6409
6410 \begingroup
6411 \LWR@restoreorigformatting
6412 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```
6413 \newcommand*{\LWR@endhideamsmath}{
6414 \endgroup
6415
6416 \LWR@htmlclosecomment
6417 \LWR@orignewline
6418 \LWR@startpars
6419 }
```

63.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an \mathcal{AMS} package, and thus requires special handling.

For svg math: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWRAMSMATHBODY` argument telling `lateximage` to use as the HTML ALT tag the environment's contents which were automatically captured by the \mathcal{AMS} environment.

For MathJax: Each environment is syched with $\mathbb{E}X$'s equation numbers, typeset with $\mathbb{E}X$ inside an HTML comment, then printed to HTML output for MathJax to process.

Env `eqnarray` This environmnet is not an \mathcal{AMS} environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into `\BODY`.

```
6420 \let\LWR@origeqnarray\eqnarray
6421 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6422 \newbool{LWR@numbereqnarray}
6423 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
6424 \newcommand{\LWR@eqnarrayfactor}{%
```

Different behavior depending on MathJax vs svg math:

```
6425 \ifbool{mathjax}
6426 {
```

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:

```
6427 \LWR@syncmathjax
6428 \boolfalse{LWR@amsmultline}
6429 \ifbool{LWR@numbereqnarray}
6430 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
6431 \LWR@beginhideamsmath
6432 \LWR@origeqnarray
6433 \BODY
6434 \LWR@origendeqnarray
6435 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
6436      \LWR@addmathjax{eqnarray}{\BODY}
6437    }%
6438    {% not LWR@numbreqnarray
```

If not numbering equations, just create the contents for MathJax:

```
6439      \LWR@addmathjax{eqnarray*}{\BODY}
6440    }% LWR@numbreqnarray
6441 }% mathjax
6442 {% not mathjax
6443      \ifbool{LWR@numbreqnarray}
6444    {
```

For numbered svg equations, first create a lateximage with an alt attribute containing sanitized copy of the source code:

```
6445      \begin{lateximage}[\LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual eqnarray:

```
6446      \LWR@origeqnarray
6447      \BODY
6448      \LWR@origendeqnarray
6449    \end{lateximage}
6450  }%
6451  {% not LWR@numbreqnarray
```

If not numbered, do the same, but an extra \nonumber seems to be required:

```
6452      \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
6453      \LWR@origeqnarray
6454      \BODY
6455      \nonumber
6456      \LWR@origendeqnarray
6457    \end{lateximage}
6458  }% LWR@numbreqnarray
6459 }% not mathjax
```

Default to number equations in the future:

```
6460 \booltrue{LWR@numbreqnarray}
6461 }
```

eqnarray itself is made with a blank line before and after to force it to be on its own line:

```

6462 \RenewEnviron{eqnarray}
6463 {%
6464
6465 \LWR@eqnarrayfactor
6466
6467 }

```

The starred version is patched to turn off the numbering:

```

6468 \csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

The following \mathcal{AMS} environments are more easily patched in-place:

Env `multline`

```

6469 \BeforeBeginEnvironment{multline}{
6470
6471 \ifbool{mathjax}
6472 {
6473     \LWR@syncmathjax
6474     \booltrue{LWR@amsmultline}
6475     \LWR@beginhideamsmath
6476 }
6477 {
6478     \lateximage[LWRAMSMATHBODY]
6479 }
6480 }
6481
6482 \AfterEndEnvironment{multline}{
6483
6484 \ifbool{mathjax}
6485 {
6486     \LWR@endhideamsmath
6487     \boolfalse{LWR@amsmultline}
6488     \LWR@addmathjax{multline}{\the\@envbody}
6489 }
6490 {\endlateximage}
6491
6492 }

```

Env `multline*`

```

6493 \BeforeBeginEnvironment{multline*}{
6494
6495 \ifbool{mathjax}
6496 {
6497     \LWR@syncmathjax

```

```

6498     \booltrue{LWR@amsmultline}
6499     \LWR@beginhideamsmath
6500 }
6501 {
6502     \lateximage[LWRAMSMATHBODY]
6503 }
6504 }
6505
6506 \AfterEndEnvironment{multline*}{
6507
6508 \ifbool{mathjax}
6509 {
6510     \LWR@endhideamsmath
6511     \boolfalse{LWR@amsmultline}
6512     \LWR@addmathjax{multline*}{\the\@envbody}
6513 }
6514 {\endlateximage}
6515
6516 }
6517

```

Env **gather**

```

6518 \BeforeBeginEnvironment{gather}{
6519
6520 \ifbool{mathjax}
6521 {
6522     \LWR@syncmathjax
6523     \boolfalse{LWR@amsmultline}
6524     \LWR@beginhideamsmath
6525 }
6526 {
6527     \lateximage[LWRAMSMATHBODY]
6528 }
6529 }
6530
6531 \AfterEndEnvironment{gather}{
6532
6533 \ifbool{mathjax}
6534 {
6535     \LWR@endhideamsmath
6536     \LWR@addmathjax{gather}{\the\@envbody}
6537 }
6538 {\endlateximage}
6539
6540 }

```

Env **gather***

```

6541 \BeforeBeginEnvironment{gather*}{
6542
6543 \ifbool{mathjax}
6544 {
6545     \LWR@syncmathjax
6546     \boolfalse{LWR@amsmultline}
6547     \LWR@beginhideamsmath
6548 }
6549 {
6550     \lateximage[LWRAMSMATHBODY]
6551 }
6552 }
6553
6554 \AfterEndEnvironment{gather*}{
6555
6556 \ifbool{mathjax}
6557 {
6558     \LWR@endhideamsmath
6559     \LWR@addmathjax{gather*}{\the\@envbody}
6560 }
6561 {\endlateximage}
6562
6563 }
```

Env **align**

```

6564 \BeforeBeginEnvironment{align}{
6565
6566 \ifbool{mathjax}
6567 {
6568     \LWR@syncmathjax
6569     \boolfalse{LWR@amsmultline}
6570     \LWR@beginhideamsmath
6571 }
6572 {
6573     \lateximage[LWRAMSMATHBODY]
6574 }
6575 }
6576
6577 \AfterEndEnvironment{align}{
6578
6579 \ifbool{mathjax}
6580 {
6581     \LWR@endhideamsmath
6582     \LWR@addmathjax{align}{\the\@envbody}
6583 }
6584 {\endlateximage}
```

```

6585
6586 }

```

Env **align***

```

6587 \BeforeBeginEnvironment{align*}{
6588
6589 \ifbool{mathjax}
6590 {
6591     \LWR@syncmathjax
6592     \boolfalse{LWR@amsmultline}
6593     \LWR@beginhideamsmath
6594 }
6595 {
6596     \lateximage[LWR@SMATHBODY]
6597 }
6598 }
6599
6600 \AfterEndEnvironment{align*}{
6601
6602 \ifbool{mathjax}
6603 {
6604     \LWR@endhideamsmath
6605     \LWR@addmathjax{align*}{\the\@envbody}
6606 }
6607 {\endlateximage}
6608
6609 }

```

Env **flalign**

```

6610 \BeforeBeginEnvironment{flalign}{
6611
6612 \ifbool{mathjax}
6613 {
6614     \LWR@syncmathjax
6615     \boolfalse{LWR@amsmultline}
6616     \LWR@beginhideamsmath
6617 }
6618 {
6619     \lateximage[LWR@SMATHBODY]
6620 }
6621 }
6622
6623 \AfterEndEnvironment{flalign}{
6624
6625 \ifbool{mathjax}
6626 {

```



```

6627 \LWR@endhideamsmath
6628 \LWR@addmathjax{flalign}{\the\@envbody}
6629 }
6630 {\endlateximage}
6631
6632 }

```

Env `flalign*`

```

6633 \BeforeBeginEnvironment{flalign*}{
6634
6635 \ifbool{mathjax}
6636 {
6637 \LWR@syncmathjax
6638 \boolfalse{LWR@amsmultline}
6639 \LWR@beginhideamsmath
6640 }
6641 {
6642 \lateximage[LWRAMSMATHBODY]
6643 }
6644 }
6645
6646 \AfterEndEnvironment{flalign*}{
6647
6648 \ifbool{mathjax}
6649 {
6650 \LWR@endhideamsmath
6651 \LWR@addmathjax{flalign*}{\the\@envbody}
6652 }
6653 {\endlateximage}
6654
6655 }

6656 \end{warpHTML}

```

64 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML span is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A \LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

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}`

for HTML output: 6657 `\begin{warpHTML}`

Ctr `LWR@lateximagenumber` Sequence the images.

```
6658 \newcounter{LWR@lateximagenumber}
6659 \setcounter{LWR@lateximagenumber}{0}
```

Ctr `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
6660 \newcounter{LWR@lateximagedepth}
6661 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
6662 \edef\LWR@hashmark{\string#} % for use in \write
6663 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr `LWR@LIPage` Used to reference the PDF page number of a lateximage to be written into `lateximages.txt`.

```
6664 \newcounter{LWR@LIPage}
```

```
6665 \end{warpHTML}
```

for HTML & PRINT: 6666 `\begin{warpall}`

`\LateximageFontSizeName` Declares how large to write text in the `\lateximage`. The `.svg` file text size should blend well with the surrounding HTML text size.

Do not include the leading backslash in the name.

```
6667 \newcommand*{\LateximageFontSizeName}{large}
```

```
6668 \end{warpall}
```

for HTML output: 6669 `\begin{warpHTML}`

`\LWR@HTMLsanitize` `{\langle text \rangle}`

Math expressions are converted to lateximages, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML ALT tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```

6670 \newcommand{\LWR@HTMLsanitize}[1]{%
6671 \begingroup%
6672 \LWR@FBcancel%
6673 \protect\StrSubstitute{\detokenize{#1}}{%
6674 {\detokenize{&}}%
6675 {\detokenize{&}}%[\LWR@strresult]%
6676 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
6677 {\detokenize{<}}%
6678 {\detokenize{&lt;}}%
6679 [\LWR@strresult]%
6680 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
6681 {\detokenize{>}}%
6682 {\detokenize{&gt;}}%
6683 [\LWR@strresult]%
6684 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
6685 {\detokenize{##}}%
6686 {\detokenize{#}}%
6687 [\LWR@strresult]%
6688 \LWR@strresult%
6689 \endgroup%
6690 }
```

`\LWR@HTMLsanitizeexpand` `{\langle text \rangle}`

This version expands the argument before sanitizing it.

```

6691 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
6692 \begingroup%
6693 \LWR@FBcancel%
6694 \protect\StrSubstitute{\detokenize\expandafter{#1}}{%
6695 {\detokenize{&}}%
6696 {\detokenize{&}}%
6697 [\LWR@strresult]%
6698 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
6699 {\detokenize{<}}%
6700 {\detokenize{&lt;}}%
6701 [\LWR@strresult]%
6702 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
6703 {\detokenize{>}}%
6704 {\detokenize{&gt;}}%
6705 [\LWR@strresult]%
6706 \LWR@strresult%
6707 \endgroup%

```

6708 }

Env `lateximage` [*<alttag>*]

```
6709 \catcode'\$=\active%
6710
6711 \NewDocumentEnvironment{lateximage}{0{image}}
6712 {%
6713 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
6714 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6715 {%
6716   \addtocounter{LWR@lateximagedepth}{1}%
6717 }%
```

Otherwise, this is the outer-most lateximage:

```
6718 {% start of outer-most lateximage
```

Starting a new lateximage:

```
6719   \addtocounter{LWR@lateximagenumber}{1}%
6720   \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
6721   \boolfalse{mathjax}
```

Be sure that are doing a paragraph:

```
6722   \LWR@ensuredoingapar%
```

Next file:

```
6723   \addtocounter{LWR@externalfilecnt}{1}%
6724   \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
6725   \setcounterpageref{LWR@LIpage}{\theLWR@lateximagenumber}%
6726   \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:

```
6727 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
6728 class="lateximagesource"{}} \LWR@orignewline
```

Write instructions to the `lateximages.txt` file:

```
6729 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
6730 \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`.

```
6731 \LWR@traceinfo{lateximage: about to create open comment}%
6732 \LWR@htmlopencomment%
```

One level deeper:

```
6733 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
6734 \LWR@traceinfo{lateximage: about to create new page}%
6735 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
6736 \LWR@traceinfo{lateximage: about to create minipage}%
6737 \LWR@origminipage{6in}%
6738 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a `lateximage`.

```
6739 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
6740 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
6741 \def\@mpfn{footnote}%
6742 \def\thempfn{\thefootnote}%
6743 \let\@footnotetext\LWR@footnotetext%
```

Create the `LWRlateximage<number>` label:

```
6744 \LWR@traceinfo{lateximage: about to create label}%
6745 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
6746 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```

6747 \LetLtxMacro$\LWR@origdollar%
6748 \catcode'\$=3% math shift
6749 \LetLtxMacro\(\LWR@origopenparen%
6750 \LetLtxMacro\)\LWR@origcloseparen%
6751 \LetLtxMacro\[ \LWR@origopenbracket%
6752 \LetLtxMacro\] \LWR@origclosebracket%
6753 \let\@ensuredmath\LWR@orig@ensuredmath%
6754 }% end of outer-most lateximage
6755 \LWR@traceinfo{lateximage: finished start of environment}%
6756 }% end of \begin{lateximage}

```

When the environment closes:

```

6757 {% start of \end{lateximage}
6758 \LWR@traceinfo{lateximage: starting end of environment}%

```

Nested more than one deep?

```

6759 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

6760 {\addtocounter{LWR@lateximagedepth}{-1}}%

```

If this is the outer-most lateximage:

```

6761 {% end of outer-most lateximage

```

Finish the lateximage minipage and start a new PDF page:

```

6762 \LWR@origendminipage%
6763 \LWR@orignewpage%
6764 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```

6765 \LWR@htmlclosecomment{}\LWR@orignewline%
6766 \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.

```

6767 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%

```

```

6768    {%
6769    \LWR@subinlineimage[%
6770    \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
6771    ]%
6772    {lateximage}%
6773    {lateximages\OSPathSymbol{}}{lateximage-\theLWR@externalfilecnt}%
6774    {svg}%
6775    {}%
6776    }%
6777    {%
6778    \LWR@subinlineimage[#1]{lateximage}%
6779    {lateximages\OSPathSymbol{}}{lateximage-\theLWR@externalfilecnt}{svg}{}%
6780    }%
6781 % \LWR@orignewline% Removed to prevent extra space.

```

Be sure that are doing a paragraph:

```

6782    \LWR@ensuredoingapar%

```

Close the HTML span which has the pdf totext comment and also the link to the .svg image:

```

6783    \LWR@htmltag{/span}%
6784    \ifbool{HTMLDebugComments}{%
6785        \LWR@htmlcomment{End of lateximage}%
6786    }{}%
6787 % \LWR@orignewline% Removed to prevent extra space.

```

Undo one lateximage level:

```

6788    \addtocounter{LWR@lateximagedepth}{-1}%
6789 }% end of outer-most lateximage
6790 \LWR@traceinfo{lateximage: done}%
6791 }%
6792 \catcode'\$=3% math shift
6793 \end{warpHTML}

```

for PRINT output:

```

6794 \begin{warpprint}
6795 % \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6796 \newenvironment{lateximage}[1][{}]{\endminipage}
6797 \end{warpprint}

```

65 center, flushleft, flushright

for HTML output:

```

6798 \begin{warpHTML}

```

Env **center** Replace center functionality with CSS tags:

```
6799 \renewenvironment*{center}
6800 {
6801 \LWR@forcenewpage
6802 \BlockClass{center}
6803 }
6804 {\endBlockClass}
```

Env **flushright**

```
6805 \renewenvironment*{flushright}
6806 {
6807 \LWR@forcenewpage
6808 \BlockClass{flushright}
6809 }
6810 {\endBlockClass}
```

Env **flushleft**

```
6811 \renewenvironment*{flushleft}
6812 {
6813 \LWR@forcenewpage
6814 \BlockClass{flushleft}
6815 }
6816 {\endBlockClass}

6817 \end{warpHTML}
```

66 Pre-loaded packages

for HTML output: 6818 \begin{warpHTML}

If textcomp was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```
6819 \@ifpackageloaded{textcomp}
6820 {
6821 \LWR@origRequirePackage{lwarp-textcomp}
6822 }
6823 {}
```

If graphics or graphicx were loaded before lwarp, perhaps by xunicode, explicitly load the lwarp patches now:


```

6824 \@ifpackageloaded{graphics}
6825 {
6826 \LWR@origRequirePackage{lwarp-graphics}
6827 }
6828 {}

6829 \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: 6830 \begin{warpHTML}

Options for siunitx:

```

6831 \PassOptionsToPackage{
6832     detect-mode=true,
6833     per-mode=symbol,% fraction is not seen by pdftotext
6834 %     text-celsius = {\protect\LWRsiunitx@degree{}}C},
6835 %     text-degree = {\protect\LWRsiunitx@degree},
6836 }{siunitx}
6837

6838 \end{warpHTML}

```

68 Graphics

Pkg graphics

Pkg graphicx



.pdf image files

For `\includegraphics` with .pdf files, the user should provide a .pdf image file, and also a .svg, .png, or .jpg version of the same image. **These should be referred**



no file extension

to without a file extension:


```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```


For print output, lwarp will automatically choose the .pdf if available, other some other format otherwise. For HTML, one of the other formats is used instead.

If a .pdf file is explicitly referred to with its file extension, a link to the .pdf file will appear in the HTML output.

`\includegraphics{filename.pdf}` % creates a link in HTML

other image files For .png, .jpg, or .gif image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

 **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**

units For `\includegraphics`, avoid px and % units for width and height, or enclose them inside `warpHTML` environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area.

options `\includegraphics` accepts width and height, origin, rotate and scale, plus a new class key.

HTML class With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

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

 **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The css standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \TeX , so expect some ugly results for scaling and rotating.

for PRINT output: For print output, accept and then discard the new class key:

```
6839 \begin{warpprint}
6840 \define@key{Gin}{class}{}
6841 \end{warpprint}
```

68.1 xcolor print-mode additions

Print-mode versions of new xcolor defintions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined

`\AtBeginDocument` so that the `xcolor` originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of none, in which case only the frame is drawn, allowing the background page color to show.

for HTML & PRINT: 6842 `\begin{warpall}`

After `xparse` may have been loaded ...

6843 `\AtBeginDocument{`

... and *only* if `xcolor` was loaded:

6844 `\@ifpackageloaded{xcolor}{`
6845 `\LWR@traceinfo{patching xcolor}`

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

6846 `\LetLtxMacro\colorboxBlock\colorbox`

In HTML mode, the following is done when `xcolor` is loaded. Following is the print-mode action:

6847 `\warpprintonly{`
6848 `\LetLtxMacro\LWRprint@colorboxBlock\colorbox`
6849 `\LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox`
6850 `\LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox`
6851 `}`

`\fcolorbox` `[\langle framemodel \rangle] {\langle framecolor \rangle} [\langle boxmodel \rangle] {\langle boxcolor \rangle} {\langle text \rangle}`

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`.)

6852 `\newsavebox{\LWR@colorminipagebox}`
6853
6854 `\DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%`
6855 `\LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%`

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

6856 `\begin{lrbox}{\LWR@colorminipagebox}%`
6857 `#5%`
6858 `\end{lrbox}%`

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

```

6859 \ifthenelse{\equal{#4}{none}}%
6860 {% #4 none
6861   \LWR@traceinfo{background is none}%
6862   {% scope the \colorlet
6863     \colorlet{LWR@currentcolor}{.}%
6864     \color{#2}%
6865     \fbox{%
6866       \color{LWR@currentcolor}%
6867       \usebox{\LWR@colorminipagebox}%
6868     }% fbox
6869   }% colorlet
6870 }% #4 none
6871 {% #4 not none
6872 \LWR@traceinfo{background not none}%
6873 \IfValueTF{#1}%
6874 {%
6875   \IfValueTF{#3}%
6876   {\LWR@origprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6877   {\LWR@origprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6878 }%
6879 {% no value #1
6880   \IfValueTF{#3}%
6881   {\LWR@origprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6882   {\LWR@origprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6883 }% no value #1
6884 }% #4 not none
6885 \LWR@traceinfo{LWRprint@fcolorbox done}%
6886 }

```

`\fcolorboxBlock` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```

6887 \LetLtxMacro\LWRprint@fcolorboxBlock\LWRprint@fcolorbox

```

Env `fcolorminipage` [*1:framemodel*] [*2:framecolor*] [*3:boxmodel*] [*4:boxcolor*] [*5:align*] [*6:height*] [*7:inner-align*] [*8:width*]

In print mode, becomes a `\fcolorbox` containing a minipage:

```

6888 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
6889 {%
6890 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
6891 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
6892 \IfValueTF{#7}%
6893 {\begin{minipage}[#5][#6][#7]{#8}}%
6894 {\begin{minipage}[#5][#6][#5]{#8}}%
6895 }%
6896 {%
6897 \end{minipage}%
6898 \end{lrbox}%
6899 \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`.

```
6900 \ifthenelse{\equal{#4}{none}}%
6901 {% #4 none
6902   {% scope the \colorlet
6903     \colorlet{\LWR@currentcolor}{.}%
6904     \color{#2}%
6905     \fbox{%
6906       \color{\LWR@currentcolor}%
6907       \usebox{\LWR@colorminipagebox}%
6908     }% fbox
6909   }% colorlet
6910 }% #4 none
6911 {% #4 not none
6912   \IfValueTF{#1}%
6913   {%
6914     \IfValueTF{#3}%
6915     {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6916     {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6917   }%
6918   {% no value #1
6919     \IfValueTF{#3}%
6920     {\LWRorigprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6921     {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6922   }% no value #1
6923 }% #4 not none
6924 \LWR@traceinfo{*** finished end fcolorminipage}%
6925 }
```

`\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 `lateximage`.

```

6926 \newcommand*{\LWR@restoreorigprintxcolor}{%
6927 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
6928 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
6929 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
6930 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
6931 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
6932 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
6933 }
6934
6935 \appto{\LWR@restoreorigformatting}{%
6936 \LWR@restoreorigprintxcolor%
6937 }

```

If print mode, immediately activate the print-mode enhancements for xcolor:

```


6938 \warpprintonly{\LWR@restoreorigprintxcolor}
6939
6940 \LWR@traceinfo{xcolor patches done}
6941 }{}% xcolor loaded
6942 }% AtBeginDocument

6943 \end{warppall}

```

69 Cleveref

Pkg `cleveref` `cleveref` package is used as-is with minor patches.

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

Ex:

```

\cpageref{tab:first,tab:second}
in HTML becomes:
“pages for table 4.1 and for table 4.2”

```

See `\cpagerefFor` at page 367 to redefine the message which is printed for page number references.

loading order `cleveref` and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires `cleveref`. The user’s document may not require `cleveref`, thus the user may never explicitly load it, so during HTML output `lwarp` loads it last. If the user’s document preamble uses `cleveref` options, or functions such

as `\crefname`, then `cleveref` may be loaded in the user's preamble near the end, and `lwarp`'s additional loading of `cleveref` will have no effect.

Table 8 on page 311 shows the data structure of the label/reference system as revised by `lwarp` and `cleveref`.

A few patches allow `cleveref` to work as-is:

for HTML output: 6944 `\begin{warpHTML}`

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
6945 \AtEndPreamble{
6946 \RequirePackage{cleveref}
6947 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
6948 \AfterEndPreamble{
6949 \LWR@traceinfo{Patching cleveref.}
```

`\@@setcref` `{\<kindofref>}{\<label>}`

```
6950 \renewcommand*{\@@setcref}[2]{#1{\ref{#2}}{-}{-}}
```

`\@@setcrefrange` `{\<text>}{\<label>}{\<label>}`

```
6951 \renewcommand{\@@setcrefrange}[3]{%
6952 #1{\ref{#2}}{\ref{#3}}{-}{-}{-}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
6953 \newcommand*\cpagerefFor{for}
```

`\@@setcpageref` `{\<typeofref>}{\<label>}`, where `typeofref` is “page” or “pages”

```
6954 \renewcommand*\@@setcpageref[2]{%
6955 #1{\cpagerefFor\ \cref{#2}}{-}{-}%
6956 }
```

```
6957 \renewcommand{\@@setcpagerefrange}[3]{%
6958 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{-}}
6959 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```

6960 \let\LWR@origlabel\label
6961 \let\label\LWR@newlabel
6962 \let\LWR@origref\ref
6963 \let\ref\LWR@newref% \end{ syntax highlighting
6964 \let\LWR@origpageref\pageref
6965 \let\pageref\LWR@newpageref
6966
6967
6968
6969 \end{warpHTML}

```

70 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 6970 `\begin{warpHTML}`

Env `picture`

```


6971 \BeforeBeginEnvironment{picture}{\lateximage}
6972
6973 \AfterEndEnvironment{picture}{\endlateximage}

6974 \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 \LaTeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

for HTML output: 6975 `\begin{warpHTML}`

71.1 Counters and lengths

Ctrl `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
6976 \newcounter{LWR@minipagedepth}
6977 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
6978 \newlength{\LWR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
6979 \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 \TeX definition and modified to create a `<div>` of class `mpfootnotes`:

```

6980 \def\LWR@endminipage{%
6981     \par
6982     \unskip
6983     \ifvoid\@mpfootins\else
6984         \vskip\skip\@mpfootins
6985         \normalcolor
6986         \LWR@htmldivclass{mpfootnotes}
6987         \LWR@origmedskip
6988         \unvbox\@mpfootins
6989         \LWR@htmldivclassend{mpfootnotes}
6990     \fi
6991     \@minipagefalse
6992     \color@endgroup
6993     \egroup
6994     \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.

```

6995 \newcommand*\LWR@subminipage{%
6996 \LWR@stoppars
6997 \LWR@origminipage{6in}

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

6998 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

6999 \LWR@startpars%
7000 }

```

`\LWR@endsubminipage` Closes the subminipage.

```

7001 \newcommand*\LWR@endsubminipage{%
7002 \LWR@stoppars%
7003 \LWR@endminipage% The following empty line is required:
7004
7005 }

```

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```
7006 \newbool{LWR@minipagefullwidth}
7007 \boolfalse{LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

for HTML output: 7008 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
7009 \end{warpHTML}

for PRINT output: 7010 \begin{warpprint}
7011 \newcommand*{\minipagefullwidth}{}
7012 \end{warpprint}

for HTML output: 7013 \begin{warpHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
7014 \newbool{LWR@minipagethispar}
7015 \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.

```
7016 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
7017 {%
7018 \uselengthunit{in}%
7019 \setlength{\LWR@minipagewidth}{#4}%
7020 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
7021   \addtolength{\LWR@minipagewidth}{3em}% room for frames
7022   \setlength{\linewidth}{6in}%
7023   \setlength{\textwidth}{6in}%
7024   \setlength{\textheight}{9in}%
7025 }{}%
7026 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
7027 \addtocounter{LWR@minipagedepth}{1}%
7028 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
7029 \IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

ℒ_{TEX} 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.

7030 \LWR@stoppars%

Create the <div> tag with optional alignment style:

```

7031 \LWR@traceinfo{minipage: creating div class}%
7032 \LWR@orignewpage%
7033 \LWR@htmltag{div class="minipage" style="%
7034 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
7035 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
7036 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
7037 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
7038 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
7039 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
7040 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%
```

Print the width and optional height styles:

```

7041 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
7042 \uselengthunit{PT}%
7043 \ifbool{LWR@minipagefullwidth}%
7044 {\boolfalse{LWR@minipagefullwidth}}%
7045 {%
7046   \ifthenelse{\lengthtest{#4}=\linewidth}%
7047   {}%
7048   {width:\rndprintlength{\LWR@minipagewidth} ; }%
7049 }%
7050 \LWR@traceinfo{minipage: about to print the height}%
7051 \IfValueT{#2}{height:\rndprintlength{\LWR@minipageheight} ; }%
7052 "{}%
```

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.

```

7053
7054 \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.

```

7055 \setlength{\linewidth}{#4}% the original width
7056 \setlength{\textwidth}{6in}%
7057 \setlength{\textheight}{9in}%
```

\raggedright cancels hyphenation, which will be done by HTML instead.

```

7058 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
7059 \LWR@startpars%
7060 \LWR@traceinfo{minipage: finished starting the minipage}%
7061 }
```

End the environment with \TeX processing and closing tag:

```
7062 {%
7063 \LWR@stoppars%
7064 \LWR@endminipage% The following empty line is required:
7065
7066 \LWR@htmldivclassend{minipage}%
7067 \LWR@origvspace{1\baselineskip}% required for subcaption
7068 \addtocounter{LWR@minipagedepth}{-1}%
7069 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
7070 \global\booltrue{LWR@minipagethispar}%
7071 }
```

71.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output: `\parbox` [*<pos>*] [*<height>*] [*<inner-pos>*] [*<width>*] [*<text>*]

A parbox uses the minipage code:

```
7072 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
7073 {
7074 \LWR@traceinfo{parbox of width #4}%
7075 \begin{minipage}[#1][#2][#3]{#4}
7076 #5
7077 \end{minipage}
7078 }
```

`\makebox` (*<()>*)posn) [*<width>*] [*<pos>*] [*<text>*]

```
7079 \LetLtxMacro\LWR@origmakebox\makebox
7080
7081 \RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
7082 \IfValueTF{#2}%
7083 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```
7084   {% scope
7085   \def\LWR@align{center}%
7086   \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
7087   \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
7088   \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the width argument:

```
7089   \setlength{\LWR@tempwidth}{#2}%
```

inline-block allows width and text-align to be used in a .

```
7090   \uselengthunit{PT}%
7091   \InlineClass[%
7092       display:inline-block ; %
7093       text-align:\LWR@align\ ; %
7094       width:\rndprintlength{\LWR@tempwidth}%
7095   ]%
7096   {makebox}%
```

Without a width argument, the text is simply used inline:

```
7097   {\mbox{#4}}%
7098   }% scope
7099 }%
7100 {\mbox{#4}}%
7101 }
```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

```
7102 \LetLtxMacro\LWR@origframebox\framebox
7103
7104 \RenewDocumentCommand{\framebox}{o o m}{%
7105 \fbox{\makebox[#1][#2]{#3}}%
7106 }
```

`\LWR@forceminwidth` {*<legth>*}

Sets `\LWR@atleastonept` to be at least 1pt.

```
7107 \newlength{\LWR@atleastonept}
```

```

7108
7109 \newcommand*{\LWR@forceminwidth}[1]{%
7110 \setlength{\LWR@atleastonept}{#1}%
7111 \ifthenelse{%
7112     \lengthtest{\LWR@atleastonept>0pt}\AND%
7113     \lengthtest{\LWR@atleastonept<1pt}%
7114 }{%
7115 {\setlength{\LWR@atleastonept}{1pt}}%
7116 {}}%
7117 }

```

`\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.

```

7118 \newcommand*{\LWR@blackborderpadding}{%
7119 \uselengthunit{PT}%
7120 border:\rndprintlength{\LWR@atleastonept} solid black ; %
7121 padding:\rndprintlength{\fboxsep}%
7122 }

```

`\fbox` $\{ \langle text \rangle \}$

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
7123 \let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```

7124 \newcommand{\LWRhtml@fbox}[1]{%
7125 \LWR@traceinfo{HTML fbox}
7126 \LWR@forceminwidth{\fboxrule}%
7127 \InlineClass[%
7128 \LWR@blackborderpadding%
7129 ]{fbox}{#1}
7130 }

```

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:

```
7131 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` $\{ \langle text \rangle \}$ Creates a framed HTML <div> of the text.

A print-output version is also supplied below.

```

7132 \newcommand{\fboxBlock}[1]{%
7133 \LWR@forceminwidth{\fboxrule}%
7134 \begin{BlockClass}[%
7135 \LWR@blackborderpadding%
7136 ]{\fboxBlock}
7137 #1
7138 \end{BlockClass}
7139 }

```

Env `fminipage` [*align*] [*height*] [*align*] [*width*]

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```

7140 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
7141 {%
7142 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
7143 \LWR@forceminwidth{\fboxrule}%
7144 \setlength{\LWR@tempwidth}{#4}%
7145 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
7146 \begin{BlockClass}[%
7147 \LWR@blackborderpadding ; %
7148 \uselengthunit{PT}%
7149 \IfValueT{#2}{height:\rndprintlength{\LWR@tempheight} ; }%
7150 width:\rndprintlength{\LWR@tempwidth}%
7151 ]{fminipage}%
7152 }
7153 {%
7154 \end{BlockClass}%
7155 \LWR@traceinfo{fminipage done}%
7156 }

```

`\raisebox` [*raiselen*] [*height*] [*depth*] [*text*]

```

7157 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
7158
7159 \RenewDocumentCommand{\raisebox}{m o o m}{%
7160 #4%
7161 }

7162 \end{warpHTML}

```

for HTML & PRINT: 7163 \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.

```
7164 \newsavebox{\LWR@fminipagebox}
7165
7166 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
7167 {%
```

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:

```
7168 \IfValueTF{#3}%
7169 {\def\LWR@thisalign{#3}}
7170 {\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.

```
7171 \IfValueTF{#2}%
7172 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
7173 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
7174 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
7175 \IfValueTF{#2}%
7176 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
7177 {\minipage[#1]{#4}}%
7178 }
7179 {%
```

Close the inner minipage and the LR box with the contents:

```
7180 \endminipage%
7181 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
7182 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
7183 \endminipage%
7184 }
```

```
7185 \end{warppall}
```

for PRINT output: 7186 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock  {\langle text \rangle}
```

Creates a framed HTML <div> around the text.

```
7187 \let \fboxBlock \fbox
```

Env fminipage [*align*] [*height*] [*align*] {\i>width}

Creates a frame around its contents.

```
7188 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}
```

```
7189 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}
```

```
7190 \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: 7191 \begin{warppHTML}

```
\emph  {\langle text \rangle}
```

```
7192 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
```

```
7193 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
```

```
7194 \LetLtxMacro{\emph}{\LWR@HTMLemph}
```

```
\textmd  {\langle text \rangle}
```

```
7195 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{\LWR@htmlspanclass{textmd}{#1}}
```

```
7196 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
```

```
7197 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

`\textbf` $\{\langle text \rangle\}$

```
7198 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
7199 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
7200 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm` $\{\langle text \rangle\}$

```
7201 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{\LWR@htmlspanclass{textrm}{#1}}
7202 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
7203 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf` $\{\langle text \rangle\}$

```
7204 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{\LWR@htmlspanclass{textsf}{#1}}
7205 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
7206 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt` $\{\langle text \rangle\}$

```
7207 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
7208 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
7209 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup` $\{\langle text \rangle\}$

```
7210 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{\LWR@htmlspanclass{textup}{#1}}
7211 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
7212 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit` $\{\langle text \rangle\}$

```
7213 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
7214 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7215 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc` $\{\langle text \rangle\}$

```
7216 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{\LWR@htmlspanclass{textsc}{#1}}
7217 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7218 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

`\textsl` $\{\langle text \rangle\}$

```

7219 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{\LWR@htmlspanclass{testsl}{#1}}
7220 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7221 \LetLtxMacro{textsl}{\LWR@HTMLtextsl}

```

`\textnormal` $\langle text \rangle$

```

7222 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7223 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7224 \LetLtxMacro{textnormal}{\LWR@HTMLtextnormal}

```

```

7225 \DeclareRobustCommand{\LWR@nullrmfamily}{%}
7226 \DeclareRobustCommand{\LWR@nullsffamily}{%}
7227 \DeclareRobustCommand{\LWR@nullttfamily}{%}
7228 \DeclareRobustCommand{\LWR@nullbfseries}{%}
7229 \DeclareRobustCommand{\LWR@nullmdseries}{%}
7230 \DeclareRobustCommand{\LWR@nullupshape}{%}
7231 \DeclareRobustCommand{\LWR@nullslshape}{%}
7232 \DeclareRobustCommand{\LWR@nullscshape}{%}
7233 \DeclareRobustCommand{\LWR@nullitshape}{%}
7234 \DeclareRobustCommand{\LWR@nullem}[1]{%}
7235 \DeclareRobustCommand{\LWR@nullnormalfont}{%}

```

`\LWR@nullfonts` Removes formatting during filename operations.

```

7236 \newcommand*{\LWR@nullfonts}{%}
7237 \LetLtxMacro{\emph}{\LWR@nullemph}%
7238 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7239 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7240 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7241 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
7242 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7243 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7244 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7245 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7246 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7247 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
7248 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7249 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7250 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7251 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7252 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7253 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7254 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7255 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7256 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7257 \LetLtxMacro{\em}{\LWR@nullem}%
7258 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7259 \renewcommand*{\HTMLunicode}[1]{%}

```

```
7260 \renewcommand*{\HTMLentity}[1]{}%
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
7261 \renewcommand*{\&}{and}%
7262 \renewcommand{\textsuperscript}[1]{##1}%
7263 \renewcommand{\textsubscript}[1]{##1}%
7264 \LetLtxMacro\underline\LWR@origunderline%
7265 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7266 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
7267 }
```

`\mdseries`

```
7268 \renewcommand*{\mdseries}{}
```

`\bfseries`

```
7269 \renewcommand*{\bfseries}{}
```

`\rmfamily`

```
7270 \renewcommand*{\rmfamily}{}
```

`\sffamily`

```
7271 \renewcommand*{\sffamily}{}
```

`\ttfamily`

```
7272 \renewcommand*{\ttfamily}{}
```

`\upshape`

```
7273 \renewcommand*{\upshape}{}
```

`\itshape`

```
7274 \renewcommand*{\itshape}{}
```

`\scshape`

```
7275 \renewcommand*{\scshape}{}
```

`\normalfont`

```
7276 \renewcommand*{\normalfont}{}

```

`\sp` $\{\langle text \rangle\}$

For siunitx. Must work in math mode.

```
7277 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}{}}

```

`\sb` $\{\langle text \rangle\}$

For siunitx. Must work in math mode.

```
7278 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}{}}

```

`\textsuperscript` $\{\langle text \rangle\}$

```
7279 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}

```

`\textsubscript` $\{\langle text \rangle\}$

```
7280 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}

```

`\up` $\{\langle text \rangle\}$ Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7281 \AtBeginDocument{\let\up\textsuperscript}

```

`\fup` $\{\langle text \rangle\}$ Prints superscript.

Supports `fmtcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7282 \AtBeginDocument{\let\fup\textsuperscript}

```

`\underline` $\{\langle text \rangle\}$

```
7283 \renewcommand{\underline}[1]{\InlineClass{underline}{#1}}

```

`\hfill`

```
7284 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
7285 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
7286 \renewcommand*{\dotfill}{\dots}
```

```
7287 \end{warpHTML}
```

73 Skips, spaces, font sizes

for HTML output: 7288 \begin{warpHTML}

`\`, must be redefined after `\RequirePackage{printlen}`

```
7289 \let\LWR@origcomma\,
7290 \let\LWR@origtilde~
7291 \let\LWR@origenskip\enskip
7292 \let\LWR@origquad\quad
7293 \let\LWR@origqqquad\qqquad
7294 \let\LWR@orighspace\hspace
7295 \let\LWR@origvspace\vspace
7296 \let\LWR@origrule\rule
7297 \let\LWR@origmedskip\medskip
7298 \let\LWR@origtextellipsis\textellipsis
```

Direct-formatting space commands become HTML entities:

```
7299 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
```

```
7300
7301 \renewcommand*{~}{\HTMLentity{nbsp}}
7302
7303 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
7304 \let\LWR@orignormalsize\normalsize
7305 \let\LWR@origsmall\small
```

```

7306 \let\LWR@origfootnotesize\footnotesize
7307 \let\LWR@origscriptsize\scriptsize
7308 \let\LWR@origtiny\tiny
7309 \let\LWR@origlarge\large
7310 \let\LWR@origLarge\Large
7311 \let\LWR@origLARGE\LARGE
7312 \let\LWR@orighuge\huge
7313 \let\LWR@origHuge\Huge
7314 \renewcommand*{\normalsize}{}
7315 \renewcommand*{\small}{}
7316 \renewcommand*{\footnotesize}{}
7317 \renewcommand*{\scriptsize}{}
7318 \renewcommand*{\tiny}{}
7319 \renewcommand*{\large}{}
7320 \renewcommand*{\Large}{}
7321 \renewcommand*{\LARGE}{}
7322 \renewcommand*{\huge}{}
7323 \renewcommand*{\Huge}{}
7324
7325 \renewcommand*{\onecolumn}{}
7326
7327 \renewcommand{\twocolumn}[1][]{
7328
7329 #1
7330
7331 }

```

`\newline` Uses the HTML `
` element.

```

7332 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7333 \let\n newline\LWR@newlinebr

```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` * [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside `tabular`, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```

7334 \LetLtxMacro\LWR@origendoffline\\
7335 \NewDocumentCommand{\LWR@endoffline}{s o}
7336 {%
7337 \newline%
7338 }

```

`\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, lwarp 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. lwarp tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around \hspace, \enskip, \quad, and \qquad 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 \qquad's HTML output.

```
7339 \newcommand*{\LWR@minipagestartpars}{%
7340 \ifbool{\LWR@minipagethispar}{\LWR@startpars}{}%
7341 }
```

\LWR@minipagestoppars Placed just after \hspace, \quad, or \qquad's HTML output.

```
7342 \newcommand*{\LWR@minipagestoppars}{%
7343 \ifbool{\LWR@minipagethispar}{\LWR@stoppars}{}%
7344 }
```

\quad Handles special minipage & horizontal space interactions.

```
7345 \renewcommand*{\quad}{%
7346 \LWR@minipagestoppars%
7347 \HTMLUnicode{2001}%
7348 \LWR@minipagestartpars%
7349 }
```

\qquad Handles special minipage & horizontal space interactions.

```
7350 \renewcommand*{\qquad}{\quad\quad}
```

\enskip Handles special minipage & horizontal space interactions.

```
7351 \renewcommand*{\enskip}{%
7352 \LWR@minipagestoppars%
7353 \HTMLUnicode{2000}%
7354 \LWR@minipagestartpars%
7355 }
```

```

Len  \WR@tempwidth  Used to compute span width, height, raise for \hspace and \rule:
Len  \WR@tempheight 7356 \newlength{\LWR@tempwidth}
Len  \WR@tempraise  7357 \newlength{\LWR@tempheight}
                   7358 \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 `\qqquad`.

```

7359 \NewDocumentCommand{\LWR@hspace}{s m}{%
7360 \setlength{\LWR@tempwidth}{#2}%

```

If `\fill`, change to `\qqquad`:

```

7361 \ifnum\gluestretchorder\LWR@tempwidth>0%
7362 \setlength{\LWR@tempwidth}{2em}%
7363 \fi%

```

Only if the width is not zero:

```
7364 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{Opt}}{ }{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7365 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```

7366 \ifthenelse{\dimtest{\LWR@tempwidth}{=}{.16667em}}
7367 {%
7368 \HTMLUnicode{2009}% thin breakable space
7369 }%

```

Print the span with the converted width. Not rounded.

```

7370 {%
7371 \uselengthunit{PT}%
7372 \LWR@htmltagc{%
7373 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
7374 }%
7375 \LWR@htmltagc{/span}%
7376 }%

```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7377 \LWR@minipagestartpars%
7378 }% width not 0
7379 }
```

`\LWR@nohspace` * $\langle length \rangle$

Used to disable `\hspace` while creating description `\items`.

```
7380 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
7381 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` * $\langle length \rangle$ Nullified `vspace`.

```
7382 \NewDocumentCommand{\LWR@vspace}{s m}{}%
```

`\vspace` * $\langle length \rangle$ Nullified.

```
7383 \let\vspace\LWR@vspace
```

`\linebreak` [$\langle num \rangle$] Inserts an HTML `br` tag.

```
7384 \renewcommand*{\linebreak}[1] [] {\newline}
```

`\nolinebreak` [$\langle num \rangle$]

```
7385 \renewcommand*{\nolinebreak}[1] [] {}%
```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
7386 \renewcommand*{\pagebreak}[1] [] {
7387
7388 }
```

`\nopagebreak` [$\langle num \rangle$]

```
7389 \renewcommand*{\nopagebreak}[1] [] {}%
```

`\enlargethispage` * $\langle len \rangle$

```
7390 \RenewDocumentCommand{\enlargethispage}{s m}{}

```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
7391 \newcommand*{\LWR@currenttextcolor}{black}

```

`\LWR@rule` [$\langle raise \rangle$] $\langle width \rangle$ $\langle height \rangle$

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.

```
7392 \NewDocumentCommand{\LWR@rule}{o m m}{%

```

The width is copied into a temporary \TeX length, from which comparisons and conversions may be made:

```
7393 \setlength{\LWR@tempwidth}{#2}%

```

If it's zero-width then skip the entire rule:

```
7394 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
7395 {}% zero- width
7396 {% non-zero width

```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
7397 \ifthenelse{%
7398 \lengthtest{\LWR@tempwidth>0pt}\AND%
7399 \lengthtest{\LWR@tempwidth<1pt}%
7400 }%
7401 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```
7402 \setlength{\LWR@tempheight}{#3}%
7403 \ifthenelse{%
7404 \lengthtest{\LWR@tempheight>0pt}\AND%
7405 \lengthtest{\LWR@tempheight<1pt}%
7406 }%
7407 {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
7408 \LWR@minipagestoppars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in \LaTeX code.

```
7409 \uselengthunit{PT}%
7410 \LWR@htmltagc{%
7411 span
7412 style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
7413 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
7414 width:\printlength{\LWR@tempwidth}; %
7415 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 \LaTeX length allows a typical \LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of \LaTeX ex height, then print that to HTML with an ex unit.

```
7416 \IfValueT{#1}%
7417 {%
7418 \setlength{\LWR@tempraise}{Opt-#1}%
7419 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7420 \LWR@orignewline%
7421 -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7422 \LWR@orignewline%
7423 -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
7424 \LWR@orignewline%
7425 transform: translate(0pt,\printlength{\LWR@tempraise}); %
7426 \LWR@orignewline%
7427 }%
```

Display inline-block to place the span inline with the text:

```
7428 display:inline-block; "%
7429 }%
7430 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7431 \LWR@minipagestartpars%
7432 }% non-zero width
7433 }
```

```
\rule [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Handles special minipage & horizontal space interactions.

```
7434 \renewcommand{\rule}{\LWR@rule}
```

```
7435 \end{warpHTML}
```

74 \phantomsection

for HTML output: 7436 \begin{warpHTML}

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into table of contents:

```
7437 \newcommand*{\phantomsection}{%
7438 \section*{}%
7439 }
```

```
7440 \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: 7441 `\begin{warpHTML}`

`\TeX` $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`latexlogo` is a CSS class used to properly typeset the E and A in $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$, $\mathrm{ConT}_{\mathrm{E}}\mathrm{X}$ t, etc.

```
7442 \let\LWR@origTeX\TeX
7443
7444 \newcommand*{\LWR@TeX}
7445 {\InlineClass{latexlogofont}%
7446 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

`\LaTeX` $\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}2_{\varepsilon}$

`\LaTeXe`

```
7447 \let\LWR@origLaTeX\LaTeX
7448
7449 \newcommand*{\LWR@LaTeX}
7450 {\InlineClass{latexlogofont}%
7451 {\InlineClass{latexlogo}%
7452 {L\textsuperscript{a}T\textsubscript{e}X}}}}
7453
7454 \let\LWR@origLaTeXe\LaTeXe
7455
7456 \renewcommand*{\LaTeXe}
7457 {\LaTeX\InlineClass{latexlogofont}%
7458 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

`\LuaTeX` $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$, $\mathrm{LuaL}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`\LuaLaTeX`

```
7459 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
7460 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

`\XeTeX` $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{X}_{\mathrm{E}}\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`\XeLaTeX`

`xetexlogo` is a CSS class which aligns the backwards E in $\mathrm{X}_{\mathrm{E}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and spaces $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in $\mathrm{X}_{\mathrm{E}}\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and spaces $\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ appropriately.

```
7461 \newcommand*{\Xe}
```

```

7462 {X\textsubscript{\HTMLunicode{18e}}}
7463 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
7464 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}

```

`\ConTeXt` `ConTEXt`

```

7465 \newcommand*{\LWR@ConTeXt}
7466 {\InlineClass{latexlogofont}{Con}\TeX}%
7467 \InlineClass{latexlogofont}{t}}

```

`\BibTeX` `BIBTEX`, *MakeIndex*
`\MakeIndex`

```

7468 \providecommand*{\BibTeX}
7469 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
7470
7471 \newcommand*{\MakeIndex}
7472 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

`\AmS` *AMS*

`amslogo` is a CSS class used for the *AMS* logo.

```

7473 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
7474 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}

```

`\MiKTeX` `MiKTEX`

```

7475 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}

```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the *LyX* logo.

```

7476 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}

```

```

7477 \end{warpHTML}

```

75.2 Print logos

for PRINT output:

```

7478 \begin{warpprint}
7479 \newcommand*{\XeTeXrevE}
7480 {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
7481 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
7482 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}

```



```

7483 \providecommand*\AmS}{%
7484 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
7485 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
7486 \newcommand*\LyX{\textsf{LyX}}
7487 \providecommand*\LuaTeX{\mbox{Lua\TeX}}
7488 \providecommand*\LuaLaTeX{\mbox{Lua\LaTeX}}
7489 \providecommand*\BibTeX{\mbox{B\textsc{ib}\TeX}}
7490 \providecommand*\MakeIndex{\mbox{\textit{MakeIndex}}}
7491 \providecommand*\ConTeXt{\mbox{Con\TeXt}}
7492 \providecommand*\MiKTeX{\mbox{MiK\TeX}}
7493 \end{warpprint}

```

76 \AtBeginDocument, \AtEndDocument

for HTML output: 7494 \begin{warppHTML}

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.

`\LWR@LwarpEnd`

```
7495 \AfterEndPreamble{\LWR@LwarpStart}
```

```
7496 \AtEndDocument{\LWR@LwarpEnd}
```

```
7497 \end{warppHTML}
```

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 **lwarp-a4.sty**

§ 78 Package **a4**

Pkg a4 a4 is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4}
 2 \newcommand*{\WideMargins}{}

File 3 **lwarp-a4wide.sty**

§ 79 Package **a4wide**

Pkg a4wide a4wide is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4wide}

File 4 **lwarp-a5comb.sty**

§ 80 Package **a5comb**

Pkg a5comb a5comb is emulated.


for HTML output: 1 \LWR@ProvidesPackageDrop{a5comb}

File 5 **lwarp-abstract.sty**

§ 81 Package **abstract**

(Based on original code by PETER WILSON.)

Pkg abstract abstract is supported and patched by lwarp.

 **missing TOC** If using the number option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a

file split, which would put a table of contents out of the home page if it is after the abstract.

for HTML output: Accept all options for lwarp-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*\abstitlestitle}%
12 {\abstractnamefont%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18 \renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \@beginparpenalty\@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 \endparpenalty\@M
33 \end{\absnamepos}%
34 %% \vspace{\abstitlestitle}%
35 \fi
36 \fi
37 \vspace{\abstitlestitle}%
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 **lwarp-adjmulticol.sty**

§ 82 Package **adjmulticol**

Pkg `adjmulticol` `adjmulticol` is emulated.

Emulation similar to `multicols` is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, `lwarp` allows a variable number of columns up to three.

for HTML output: `1 \LWR@ProvidesPackageDrop{adjmulticol}`

`2 \RequirePackage{multicol}`

`adjmulticols` * `{\langle numcols \rangle}{\langle left margi \rangle}{\langle right margin \rangle}`

`3 \NewDocumentEnvironment{adjmulticols}{s m m m}`
`4 {%`

Compute the margins, and limit to positive only:

```

5 \setlength{\LWR@templengthone}{#3}%
6 \ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{}%
7 \setlength{\LWR@templengthtwo}{#4}
8 \ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{}%

```

If one column is specified, use a <div> of class singlecolumn, else use multicols:

```
9 \newcommand*\LWR@mccolstype{multicols}%
10 \ifnumcomp{#2}{=}{1}{\renewcommand*\LWR@mccolstype{singlecolumn}}{}}%
```

Help avoid page overflow:

```
11 \LWR@forcenewpage%
```

Print margins in points:

```
12 \uselengthunit{PT}%
```

Create the <div> with the given margin and class:

```
13 \BlockClass[%
14 margin-left:\rndprintlength{\LWR@templengthone} ; %
15 margin-right:\rndprintlength{\LWR@templengthtwo}%
16 ]{\LWR@mccolstype}%
17 }
18 {\endBlockClass}
```

File 7 **lwarp-addlines.sty**

\$ 83 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 8 **lwarp-afterpage.sty**

\$ 84 Package **afterpage**

Pkg afterpage Emulated.

for HTML output: Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}
```

```
2 \newcommand{\afterpage}[1]{#1}
```


File 9 lwarp-algorithmicx.sty

§ 85 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 [232.1](#).

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@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%
19 }
```

```
20 \end{warpHTML}
```

File 10 lwarp-alltt.sty

§ 86 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 11 **lwarp-amsthm.sty**

§ 87 Package **amsthm**

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

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}% new
8   }{%
9     \thm@style{#1}%

```



```

10 \renewcommand{\LWR@newtheoremstyle}{#1}% new
11 }%
12 }

```

Patched to remember the style for this theorem type:

```

13 \def\xnthm#1#2{%
14 \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
15 \let\@tempa\relax
16 \exp\@ifdefinable\csname #2\endcsname{%
17 \global\exp\let\csname end#2\endcsname\endtheorem
18 \ifx *#1% unnumbered, need to get one more mandatory arg
19 \edef\@tempa##1{%
20 \gdef\exp\@nx\csname#2\endcsname{%
21 \@nx\@thm{\exp\@nx\csname th@the\thm@style\endcsname}%
22 }{##1}}}%
23 \else % numbered theorem, need to check for optional arg
24 \def\@tempa{\@oparg{\ynthm{#2}}[]}%
25 \fi
26 \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% new
27 }%
28 \@tempa
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@\@plus6\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 12 **lwarp-anonchap.sty**

§ 88 Package **anonchap**

Pkg anonchap anonchap is emulated.

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

The code is shared by tocbibind.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{anonchap}

2 \newcommand{\simplechapter}[1][\@empty]{%
3   \def\@chapcntformat##1{%
4     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
5   }%
6 }
7
8 \newcommand{\restorechapter}{%
9 \let\@chapcntformat\@seccntformat%
10 }
```

File 13 **lwarp-anysize.sty**

\$ 89 Package **anysize**

Pkg **anysize** **anysize** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{anysize}

2 \def\papersize#1#2{}
3 \def\marginsize#1#2#3#4{}
```

File 14 **lwarp-appendix.sty**

\$ 90 Package **appendix**

Pkg **appendix** **appendix** is patched for use by lwarp.



incorrect TOC link

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

for HTML output:

```

1 \LWR@ProvidesPackagePass{appendix}

2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addappheadtotoc
6 \fi
7 }
```

```

8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addappheadtotoc
13 \fi
14 }

```

File 15 **lwarp-arabicfront.sty**

§ 91 Package **arabicfront**

Pkg arabicfront arabicfront is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{arabicfront}

File 16 **lwarp-authblk.sty**

§ 92 Package **authblk**

Pkg authblk authblk is patched for HTML.

package support lwarp supports the native \LaTeX titling commands, and also supports the packages
load order 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{lwarp-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\LWRAB@affil\affil
9
10 \renewcommand{\affil}[2] [] {%
11 \LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
12 }

```

Create an HTML break for an `\authorcr`:

```

13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}

```

File 17 `lwarp-balance.sty`

§ 93 Package **balance**

Pkg `balance` Emulated.

for HTML output: Discard all options for `lwarp-balance`:

```

1 \LWR@ProvidesPackageDrop{balance}

2 \newcommand*{\balance}{}
3 \newcommand*{\nobalance}{}


```

File 18 `lwarp-bigdelim.sty`

§ 94 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%`

 **use `\mrowcell`** `\ldelim` and `\rdelim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\ldelim` or `\rdelim`, but not in cells which are above or below the delimiter:

```

\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}

```

$$\begin{array}{cc} & a & b \\ \text{left} \left\{ & c & d \\ & e & f \\ & g & h \end{array}$$

for HTML output:

First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case `bigdelim` was not loaded:

```

1 \let\ldelim\relax
2 \let\rdelim\relax

```

Next, load the package's new definitions:

```

3 \LWR@ProvidesPackagePass{bigdelim}

```

Remember the print-mode versions:

```

4 \LetLtxMacro\LWR@origldelim\ldelim
5 \LetLtxMacro\LWR@origrdelim\rdelim

```

```

\ldelim {\langle 1:delimiter \rangle} {\langle 2:#rows \rangle} {\langle 3:width \rangle} [\langle 4:text \rangle]
\rdelim

6 \RenewDocumentCommand{\ldelim}{m m m O{}}{%
7 \renewcommand{\LWR@multirowborder}{right}%
8 \multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
9 }
10
11 \RenewDocumentCommand{\rdelim}{m m m O{}}{%
12 \renewcommand{\LWR@multirowborder}{left}%
13 \multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
14 }

```

When entering a `lateximage`, restore the print-mode versions:

```

15 \appto{\LWR@restoreorigformatting}{%
16 \LetLtxMacro{\ldelim}{\LWR@origldelim}%
17 \LetLtxMacro{\rdelim}{\LWR@origrdelim}%
18 }

```

File 19 **lwarp-bigstrut.sty**

§ 95 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 20 **lwarp-bookmark.sty**

§ 96 Package **bookmark**

Pkg bookmark bookmark is emulated during HTML output, and the bookmark package is ignored.

for HTML output: Discard all options for lwarp-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 21 **lwarp-booktabs.sty**

§ 97 Package **booktabs**

Pkg booktabs booktabs is emulated during HTML output, and used as-is during print output and inside an HTML lateximage.

for HTML output:

```
1 \LWR@ProvidesPackagePass{booktabs}
```

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML lateximages. 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 22 **lwarp-boxedminipage.sty**

§ 98 Package **boxedminipage**

Pkg boxedminipage boxedminipage is superceded by boxedminipage2e.

for HTML output: 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

File 23 **lwarp-boxedminipage2e.sty**

§ 99 Package **boxedminipage2e**

Pkg boxedminipage2e boxedminipage2e is emulated.

for HTML output: Discard all options for lwarp-boxedminipage2e:

```
1 \LWR@ProvidesPackageDrop{boxedminipage2e}

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}%
4 \minipage%
5 }
6 {
7 \endminipage%
8 \end{BlockClass}
9 }
```

File 24 **lwarp-caption2.sty**

§ 100 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 25 **lwarp-ccaption.sty**

§ 101 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 26 **lwarp-changepage.sty**

§ 102 Package **changepage**

Pkg changepage changepage is ignored.

for HTML output: Discard all options for lwarp-changepage:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpge
3 \DeclareRobustCommand{\checkoddpge}{\oddpgetrue}
4 \DeclareRobustCommand{\changetext}[5]{ }
5 \DeclareRobustCommand{\changepage}[9]{ }
6 \newenvironment{adjustwidth}[2]{ }{ }
7 \newenvironment{adjustwidth*}[2]{ }{ }
```

File 27 **lwarp-chngpage.sty**

§ 103 Package **chngpage**

Pkg chngpage chngpage is superceded by changepage.

for HTML output: `1 \LWR@loadnever{chngepage}{change page}`

File 28 **lwarp-chappg.sty**

\$ 104 Package **chappg**

Pkg chappg chappg is ignored.


for HTML output: `1 \LWR@ProvidesPackageDrop{chappg}`
`2 \renewcommand{\pagenumbering}[2] [] {}`
`3 \providecommand{\chappgsep}{--}`

File 29 **lwarp-color.sty**

\$ 105 Package **color**

Pkg color Allowed but ignored. xcolor is then required as well.

color is superceded by xcolor, and lwarp requires several of the features of xcolor.

 **missing colors** It should be sufficient for the user's document to load color then load xcolor as well.

for HTML output: `1 \LWR@ProvidesPackagePass{color}`
`2 \RequirePackage{xcolor}`

File 30 **lwarp-crop.sty**

\$ 106 Package **crop**

Pkg crop Emulated.

for HTML output: Discard all options for lwarp-crop:

`1 \LWR@ProvidesPackageDrop{crop}`
`2 \newcommand*{\crop}[1] [] {}`
`3 \newcommand*{\cropdef}[6] [] {}`

File 31 **lwarp-cuted.sty**

§ 107 Package **cuted**

Pkg cuted cuted is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{cuted}

2 \newenvironment{strip}{}{}
3 \newskip\stripsep
4 \def\oldcolsbreak#1{}

File 32 **lwarp-cutwin.sty**

§ 108 Package **cutwin**

Pkg cutwin Emulated.

for HTML output: Discard all options for lwarp-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 33 **lwarp-dblfnote.sty**

§ 109 Package **dblfnote**

Pkg dblfnote dblfnote is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{dblfnote}

```

2 \newcounter{DFNsloppiness}
3 \newdimen\DFNcolumnsep
4 \newdimen\DFNcolumnwidth
5 \def\DFNallowcbreak{}
6 \def\DFNinhibitcbreak{}
7 \def\DFNtrysingle{}
8 \def\DFNalwaysdouble{}
9 \def\DFNruleboth{}
10 \def\DFNruleleft{}

```

File 34 **lwarp-dcolumn.sty**

§ 110 Package **dcolumn**

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

```

1 \LWR@ProvidesPackageDrop{dcolumn}

```

File 35 **lwarp-draftwatermark.sty**

§ 111 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 36 **lwarp-ebook.sty**

§ 112 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 37 **lwarp-ellipsis.sty**

§ 113 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 38 **lwarp-emptypage.sty**

§ 114 Package **emptypage**

Pkg **emptypage** emptypage is ignored.

for HTML output: Discard all options for lwarp-emptypage:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

File 39 **lwarp-endnotes.sty**

§ 115 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 40 **lwarp-enumerate.sty**

§ 116 Package **enumerate**

Pkg `enumerate` `enumerate` is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by lwarp and thus must no longer be dropped.

for HTML output: `1 \LWR@ProvidesPackagePass{enumerate}`

File 41 **lwarp-enumitem.sty**

§ 117 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 42 **lwarp-epigraph.sty**§ 118 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]{flushleft}
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 43 **lwarp-eso-pic.sty**

§ 119 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 44 **lwarp-everypage.sty**

§ 120 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 45 **lwarp-everyshi.sty**

§ 121 Package **everyshi**

Pkg everyshi Emulated.

for HTML output: Discard all options for lwarp-everyshi:

```
1 \LWR@ProvidesPackageDrop{everyshi}

2 \newcommand*{\EveryShipout}[1]{}
3 \newcommand*{\AtNextShipout}[1]{}

```

File 46 **lwarp-extramarks.sty**

§ 122 Package **extramarks**

Pkg extramarks extramarks is not used.

for HTML output: Discard all options for lwarp-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 47 **lwarp-fancybox.sty**

§ 123 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 `lateximage`, which is done by adding `\lateximage` at the very start of `FramedEqn`'s beginning code, and `\endlateximage` at the very end of the ending code. Unfortunately, the `HTML alt` attribute is not used here.

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

[framing alternatives](#) `\fbox` works with `fancybox`. Also see `lwarp`'s `\fboxBlock` macro and `fminipage` environment for alternatives to `\fbox` for framing environments.

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

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

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

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

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

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

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}
```

`Beqnarray` is adapted for MathJax or enclosed inside a `lateximage`:

```

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}{#3\quad#4}{#4\quad#3}%
47 }

```

Neutralized commands:

```

48 \RenewDocumentCommand{\fancyput}{s d() m}{\}
49 \RenewDocumentCommand{\thisfancyput}{s d() m}{\}
50
51 \RenewDocumentCommand{\fancypage}{m m}{\}
52 \RenewDocumentCommand{\thisfancypage}{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*{-.5\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 48 **lwarp-fancyhdr.sty**

§ 124 Package **fancyhdr**

Pkg fancyhdr fancyhdr is nullified.

for HTML output: Discard all options for lwarp-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 49 **lwarp-fancyvrb.sty**

§ 125 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{lwarp}
7 {Verbatim footnotes are not yet supported by lwarp.}
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 definition 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@single{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{}}%
89 \let\FV@BeginListFrame\LWR@fvstartsingle%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
91 \let\FV@RightListFrame\FV@RightListFrame@Single%
92 \let\FV@EndListFrame\LWR@fvendsingle}
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\FV@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\FV@FrameRule}%
143 \else
144 \ifnum#1=\z@
145 % \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
146 \ifx\FV@LabelPositionTopLine\relax
147 \else
148 \LWR@FVfindbordercolor
149 \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
150 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
151 \LWR@htmltagc{/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@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
159 \LWR@origtextrm{\FV@LabelEnd}
160 \LWR@htmltagc{/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 \ifcvoid{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 50 **lwarp-figcaps.sty**

§ 126 Package **figcaps**

Pkg **figcaps** Emulated.

for HTML output: Discard all options for lwarp-figcaps:

```

1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}{}
3 \newcommand*{\figcapsoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}

```

File 51 **lwarp-fix2col.sty**

§ 127 Package **fix2col**

Pkg fix2col fix2col is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fix2col}

File 52 **lwarp-float.sty**

§ 128 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 {<1: type>} {<2: placement>} {<3: ext>} [<4: within>]

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 {<type>} {<name>}

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` $\{\langle style \rangle\}$

Float styles are ignored.

```
15 \newcommand{\floatstyle}[1]{%
16 }
```

`\restylefloat` $* \{\langle style \rangle\}$

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%
18 }
```

File 53 **lwarp-floatflt.sty**

§ 129 Package **floatflt**

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 \LWR@forcenewpage
6 \LWR@stoppars%
7 \LWR@htmltag{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{KFLTfloatflt@marginfloat}{figure}}
17 {\end{KFLTfloatflt@marginfloat}}
18
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20 {\begin{KFLTfloatflt@marginfloat}{table}#2}
21 {\end{KFLTfloatflt@marginfloat}}

```

File 54 **lwarp-floatpag.sty**

§ 130 Package **floatpag**

Pkg floatpag Emulated.

for HTML output: Discard all options for lwarp-floatpag:

```

1 \LWR@ProvidesPackageDrop{floatpag}

2 \newcommand*{\floatpagestyle}[1]{}
3 \newcommand*{\rotfloatpagestyle}[1]{}
4 \newcommand*{\thisfloatpagestyle}[1]{}


```


File 55 **lwarp-floatrow.sty**

§ 131 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. lwarp will scale the HTML lengths appropriately.

- Use `warpprint` and `warpHTML` environments to select appropriate values for each case.
- Inside a `warpHTML` environment, manually change `\FBwidth` or `\FBheight` before the `\ffigbox` or `\ttabbox`. Use `\FBwidth` or `\FBheight` normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

After everything has loaded, remember whether subcaption was loaded. If not, it is assumed that `subfig` is used instead:

```

2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }

```

`\floatbox` [*⟨1 preamble⟩*] [*⟨2 captype⟩*] [*⟨3 width⟩*] [*⟨4 height⟩*] [*⟨5 vert pos⟩*] [*⟨6 caption⟩*] [*⟨7 object⟩*]

Only parameters for `captype`, `width`, `caption`, and `object` are used.

`LWR@insubfloatrow` is true if inside a `subfloatrow` environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```

9 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For `subcaption`:

```

12 \ifbool{LWR@insubfloatrow}%
13 {% subcaption in a subfloatrow

```

`subfigure` and `subtable` environments take `width` as an argument.

```

14 \IfValueTF{#3}%
15 {\@nameuse{sub#2}{#3}}%
16 {\@nameuse{sub#2}{\linewidth}}%
17 }% subcaption in a subfloatrow
18 {% subcaption not in subfloatrow

```

`figure` and `table` environments do not take a `width` argument.

```

19 \@nameuse{#2}%
20 }% subcaption not in subfloatrow
21 #6
22
23 #7

```


End the environments:

```
24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27}% subcaption
28{% assume subfig
```

For subfig:

```
29\ifbool{LWR@insubfloatrow}%
30{% subfig in a subfloatrow
```

`\subfloat` is a macro, not an environment.

Package `subfig`'s `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument #6 contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```
31 \begingroup
32 \let\caption\@firstofone
33 \subfloat[#6]{#7}
34 \endgroup
35}% subfig in a subfloatrow
36{% subfig package, but not a subfig
```

`figure` and `table` are environments:

```
37\@nameuse{#2}
38#6
39
40#7
41\@nameuse{end#2}
42}% subfig package, but not a subfig
43}% assume subfig
44}
```

Not used:

```
45\newcommand*{\nocapbeside}{}
46\newcommand*{\capbeside}{}
47\newcommand*{\captop}{}
48\newlength{\FBwidth}
49\setlength{\FBwidth}{.3\linewidth}
50\newlength{\FBheight}
51\setlength{\FBheight}{2in}
52\newcommand*{\useFCwidth}{}
53\newcommand{\floatsetup}[2][{}]{
54\newcommand{\thisfloatsetup}[1]{
55\newcommand{\clearfloatsetup}[1]{
56\newcommand*{\killfloatstyle}{}
57}
```

`\newfloatcommand` $\{\langle 1 \text{ command} \rangle\} \{\langle 2 \text{ captype} \rangle\} [\langle 3 \text{ preamble} \rangle] [\langle 4 \text{ default width} \rangle]$

Preamble and default width are ignored.

```
57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }
```

`\renewfloatcommand` $\{\langle 1 \text{ command} \rangle\} \{\langle 2 \text{ captype} \rangle\} [\langle 3 \text{ preamble} \rangle] [\langle 4 \text{ default width} \rangle]$

Preamble and default width are ignored.

```
62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }
```

`\ffigbox` $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```
67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []
```

`\ttabbox` $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```
68 \newfloatcommand{ttabbox}{table}[\capttop] [\FBwidth]
```

`\fcapside` $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```
69 \newfloatcommand{fcapside}{figure}[\capbeside] []
```

Env `floatrow` $[\langle numfloats \rangle]$

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}{}%
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}{}%

```

\DeclareNewFloatType {<type>} {<options>}

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}{}

```

`\RawCaption` $\{\langle text \rangle\}$

To be used inside a minipage or parbox.

```

127 \newcommand{\RawCaption}[1]{#1}

```

`\floatfoot` $\{\langle text \rangle\}$

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@insubfloatrow}
134 \boolfalse{LWR@insubfloatrow}

```

Env `subfloatrow` $[\langle num_floats \rangle]$

```

135 \newenvironment*{subfloatrow}[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@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```

139 \begingroup

```

```

140 \booltrue{LWR@insubfloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@insubfloatrow}
146 }

```

File 56 **lwarp-flushend.sty**

§ 132 Package **flushend**

Pkg flushend Emulated.

for HTML output: Discard all options for lwarp-flushend:

```

1 \LWR@ProvidesPackageDrop{flushend}
2 %     \end{ma-crocode}
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*{\showcolsendrule}{}

```

File 57 **lwarp-fncychap.sty**

§ 133 Package **fncychap**

Pkg fncychap fncychap is ignored.

for HTML output: Discard all options for lwarp-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 58 **lwarp-fnpos.sty**

§ 134 Package **fnpos**

Pkg fnpos fnpos is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fnpos}

2 \newcommand*{\makeFNbottom}{}
3 \newcommand*{\makeFNmid}{}
4 \newcommand*{\makeFNbelow}{}
5 \newcommand*{\makeFNabove}{}

```

File 59 **lwarp-fontenc.sty**

§ 135 Package **fontenc**

Pkg fontenc Error if fontenc is loaded after lwarp.

Discard all options for lwarp-fontenc:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fontenc}

2 \LWR@loadbefore{fontenc}

```

File 60 **lwarp-fontspec.sty**

§ 136 Package **fontspec**

Pkg fontspec Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fontspec}
2 \LWR@loadbefore{fontspec}
```

File 61 **lwarp-footmisc.sty**

§ 137 Package **footmisc**

(Based on original code by ROBIN FAIRBAIRNS.)

Pkg footmisc footmisc is emulated during HTML output, and the footmisc package is ignored.

```
1 \LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
2 \newcommand{\footnotelayout}{}
3 \newcommand{\setfnsymbol}[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\p@ \hrule \kern2.6\p@}
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     \@xmpfootnotemark
22     {%
23       \stepcounter\@mpfn
24       \protected@xdef\@thefnmark{\thempfn}%
25       \@footnotemark
26     }%
27 }
28 \def\@xmpfootnotemark[#1]{%
29   \begingroup
30     \csname c@\@mpfn\endcsname #1\relax
31     \unrestored@protected@xdef\@thefnmark{\thempfn}%
32   \endgroup
33   \@footnotemark
34 }
```

File 62 **lwarp-footnote.sty**

§ 138 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   \beginngroup%
25   \let\@makefntext\@empty%
26   \let\@finalstrut\@gobble%
27   \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
28   \@footnotetext{\unvbox\z@}%
29   \endgroup%
30 }

```

These have been redefined, so re-\let them again:

```

31 \let\endfootnote\fn@endfntext
32 \let\endfootnotetext\endfootnote

```

File 63 **lwarp-footnotehyper.sty**

§ 139 Package **footnotehyper**

Pkg **footnotehyper** footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

for HTML output: Discard all options for lwarp-footnotehyper:

```

1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}

```

File 64 **lwarp-framed.sty**

§ 140 Package **framed**

(Based on original code by DONALD ARSENEAU.)

Pkg **framed** framed is supported and patched by lwarp.

for HTML output: Accept all options for lwarp-framed:

```

1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}% for \convertcolorspec

3
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}{}{0pt}{0pt}{0pt}{0pt}{\BODY}
61 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

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}}{}{% not empty
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}}{}{% not empty
75   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77   \textcolor{TFTitleColor}{\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 65 **lwarp-ftnright.sty**

§ 141 Package **ftnright**

Pkg ftnright ftnright is ignored.

for HTML output: Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

File 66 **lwarp-fullpage.sty**

§ 142 Package **fullpage**

Pkg fullpage Emulated.

for HTML output: Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

File 67 **lwarp-fullwidth.sty**

§ 143 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][1]{%
3 \minipagefullwidth%
4 \minipage{\linewidth}%
5 }
6 {%
7 \endminipage%
8 }
```

File 68 **lwarp-geometry.sty**

§ 144 Package **geometry**

Pkg geometry geometry is preloaded by lwarp, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for lwarp-geometry:

```

1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*{\geometry}[1]{}
3 \renewcommand*{\newgeometry}[1]{}
4 \renewcommand*{\restoregeometry}{}
5 \renewcommand*{\savegeometry}[1]{}
6 \renewcommand*{\loadgeometry}[1]{}

```

File 69 **lwarp-glossaries.sty**

§ 145 Package **glossaries**

Pkg glossaries xindy is required for glossaries.

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

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

[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 lwarp 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 lwarpmk printglossary lwarpmk has the commands lwarpmk printglossary and lwarpmk htmlglossary to process the glossaries created by glossaries using xindy.

Opt lwarpmk htmlglossary

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   \ifdefempty\@@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 70 **lwarp-graphics.sty**

§ 146 Package **graphics**

 Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackagePass{graphics}

 § 146.1 **Graphics extensions**

 \DeclareGraphicsExtensions {*\list*}

 \AtBeginDocument allow SVG files instead of PDF:

```
2 \AtBeginDocument{
3 \DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
4 \DeclareGraphicsRule{.svg}{svg}{.svg}{}
5 \DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
6 }
```

 Inside a lateximage, allow PDF instead of SVG:

```
7 \appto\LWR@restoreorigformatting{%
8 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
9 }
```

§ 146.2 Length conversions and graphics options



whitespace

A scaled image in \TeX 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.

```
10 \AtBeginDocument{
11 \renewcommand*\unitspace{}}
12 }
```

Used to store the user's selected dimensions and HTML class.

The class defaults to “inlineimage” unless changed by a `class=xyx` option.

```
13 \newlength{\LWR@igwidth}
14 \newlength{\LWR@igheight}
15 \newcommand*\LWR@igwidthstyle{}
16 \newcommand*\LWR@igheightstyle{}
17 \newcommand*\LWR@igorigin{}
18 \newcommand*\LWR@igangle{}
19 \newcommand*\LWR@igxscale{1}
20 \newcommand*\LWR@igyscale{1}
21 \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:

```
22 \define@key{igraph}{width}{%
23 \setlength{\LWR@igwidth}{#1}%
24 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
25 {%
```

Default to use the converted fixed length given:

```
26 \uselengthunit{PT}%
27 \renewcommand*\LWR@igwidthstyle{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
28 \IfEndWith{#1}{ex}%
29 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes ex
30 {}% not ex
31 \IfEndWith{#1}{em}%
32 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes em
33 {}% not em
34 \IfEndWith{#1}{\}%
35 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes percent
```



```

36    {}% not percent
37    \IfEndWith{#1}{px}%
38    {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
39    {}% not px
40 }{}% end of length > Opt
41 }

```

If an optional height was given, set an HTML style:

```

42 \define@key{igraph}{height}{%
43 \setlength{\LWR@igheight}{#1}%
44 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
45 {%

```

Default to use the converted fixed length given:

```

46    \uselengthunit{PT}%
47    \renewcommand*{\LWR@igheightstyle}{%
48    height:\rndprintlength{\LWR@igheight} %
49    }%

```

If ex or em dimensions were given, use those instead:

```

50    \IfEndWith{#1}{ex}%
51    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
52    {}% not ex
53    \IfEndWith{#1}{em}%
54    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
55    {}% not em
56    \IfEndWith{#1}{\}%
57    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
58    {}% not percent
59    \IfEndWith{#1}{px}%
60    {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
61    {}% not px
62 }{}% end of length > Opt
63 }

```

Handle origin key:

```

64 \define@key{igraph}{origin}{%
65 \renewcommand*{\LWR@igorigin}{#1}%
66 }

```

Handle angle key:

```

67 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```
68 \define@key{igraph}{class}{\renewcommand*\LWR@igclass}{#1}}
69
```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```
70 \define@key{igraph}{scale}{%
71 \renewcommand*\LWR@igxscale}{#1}%
72 \renewcommand*\LWR@igyscale}{#1}}
```

Numerous ignored keys:

```
73 \define@key{igraph}{bb}{}
74 \define@key{igraph}{bbllx}{}
75 \define@key{igraph}{bbllx}{}
76 \define@key{igraph}{bburx}{}
77 \define@key{igraph}{bbury}{}
78 \define@key{igraph}{natwidth}{}
79 \define@key{igraph}{natheight}{}
80 \define@key{igraph}{hiresbb}{}
81 \define@key{igraph}{viewport}{}
82 \define@key{igraph}{trim}{}
83 \define@key{igraph}{totalheight}{}
84 \define@key{igraph}{keepaspectratio}{}
85 \define@key{igraph}{clip}{}
86 \define@key{igraph}{draft}{}
87 \define@key{igraph}{type}{}
88 \define@key{igraph}{ext}{}
89 \define@key{igraph}{read}{}
90 \define@key{igraph}{command}{}

```

`\LWR@rotstyle` $\{\langle prefix \rangle\} \{\langle degrees \rangle\}$

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.

```
91 \newcommand*\LWR@rotstyle[2]{%
92 #1transform:rotate(-#2deg);
93 }
```

`\LWR@scaletyle` $\{\langle prefix \rangle\} \{\langle xscale \rangle\} \{\langle yscale \rangle\}$

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.

```
94 \newcommand*{\LWR@scalestyle}[3]{%
95   #1transform:scale(#2,#3);
96 }
```

§ 146.3 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
97 \newbool{LWR@infloatrow}
98 \boolfalse{LWR@infloatrow}
```

`\LWR@opacity` may be set by the transparent package. For HTML it is only used for `\includegraphics`.

```
99 \def\LWR@opacity{1}
```

`\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.

```
100 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
101 {%
102   \LWR@traceinfo{\LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
103 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
104 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
105 {%
106   \ifbool{LWR@infloatrow}%
107   {%
108     {% not in a minipage or a floatrow:
109       \setlength{\linewidth}{6in}%
110       \setlength{\textwidth}{6in}%
111       \setlength{\textheight}{9in}%
112     }%
113   }{%}
```

```

114 \renewcommand*{\Gin@setfile}[3]{%
115 \LWR@traceinfo{\Gin@setfile ##3}%
116 \xdef\LWR@parsedfilename{##3}%
117 }%
118 \Gininclude@graphics{\detokenize{#4}}%
119 \filename@parse{\LWR@parsedfilename}%

```

For correct em sizing during the width and height conversions:

```

120 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

121 \setlength{\LWR@igwidth}{0pt}%
122 \setlength{\LWR@igheight}{0pt}%
123 \renewcommand*{\LWR@igwidthstyle}{}%
124 \renewcommand*{\LWR@igheightstyle}{}%
125 \renewcommand*{\LWR@igorigin}{}%
126 \renewcommand*{\LWR@igangle}{}%
127 \renewcommand*{\LWR@igxscale}{1}%
128 \renewcommand*{\LWR@igyscale}{1}%
129 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```

130 \IfValueF{#3}{%
131 \IfValueTF{#2}%
132 {\setkeys{igraph}{#2}}%
133 {\setkeys{igraph}{}}%
134 }%

```

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.

```

135 \LWR@traceinfo{\LWR@includegraphicsb: about to create href}%
136 \href{\LWR@parsedfilename}%
137 {% start of href
138 \LWR@traceinfo{\LWR@includegraphicsb: about to LWR@htmltag}%
139 \LWR@htmltag{% start of image tags
140 img src="\LWR@parsedfilename" \LWR@orignewline%
141 \LWR@origtilde{} alt="[\filename@base]" \LWR@orignewline%

```

Only include a style tag if a width, height, angle, or scale was given:

```

142 \ifthenelse{

```

```

143 \NOT\equal{\LWR@igwidthstyle}{ } \OR
144 \NOT\equal{\LWR@igheightstyle}{ } \OR
145 \NOT\equal{\LWR@igorigin}{ } \OR
146 \NOT\equal{\LWR@igangle}{ } \OR
147 \NOT\equal{\LWR@igxscale}{1} \OR
148 \NOT\equal{\LWR@igyscale}{1}
149 }%
150 {\LWR@origtilde{ } style="%
151 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
152 {\LWR@igwidthstyle; }}%
153 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
154 {\LWR@igheightstyle; }}%
155 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
156 {\LWR@origtilde{ } transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@originewline}{ }%
157 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
158 {%
159 \LWR@rotstyle{-ms-}{\LWR@igangle}%
160 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
161 \LWR@rotstyle}{\LWR@igangle%
162 } }%
163 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
164 \NOT\equal{\LWR@igyscale}{1}}%
165 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
166 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
167 \LWR@scalestyle}{\LWR@igxscale}{\LWR@igyscale}}}%
168 %
169 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
170 {opacity:\LWR@opacity;}%
171 { }%
172 %
173 " \LWR@originewline}{ }%

```

Set the class:

```

174 \LWR@origtilde{ } class="\LWR@igclass" \LWR@originewline%
175 }% end of image tags
176 }% end of href
177 \endgroup

```

Return to small-sized output:

```

178 \LWR@origscriptsize
179 \LWR@traceinfo{\LWR@includegraphicsb done}%
180 }

```

`\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.

```
181 \LetLtxMacro\LWR@originincludegraphics\includegraphics
182
183 \AtBeginDocument{
184 \renewcommand*\includegraphics{
185 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
186 \LWR@traceinfo{includegraphics}
187 \LWR@ensuredoingapar%
188 \begingroup%
189 \catcode'\_ =12%
190 \LWR@includegraphicsb%
191 }% includegraphics
192 }% AtBeginDocument
```

`\LWR@rotboxorigin` Holds the origin key letters.

```
193 \newcommand*\LWR@rotboxorigin{}
```

`\LWR@originname` `{\langle letter \rangle}`

Given one \TeX origin key value, translate into an HTML origin word:

```
194 \newcommand*\LWR@originname[1]{%
195 \ifthenelse{\equal{#1}{t}}{top}{}%
196 \ifthenelse{\equal{#1}{b}}{bottom}{}%
197 \ifthenelse{\equal{#1}{c}}{center}{}%
198 \ifthenelse{\equal{#1}{l}}{left}{}%
199 \ifthenelse{\equal{#1}{r}}{right}{}%
200 }
```

`\LWR@originnames` `{\langle letters \rangle}`

Given one- or two-letter \TeX origin key values, translate into HTML origin words:

```
201 \newcommand*\LWR@originnames[1]{%
202 \StrChar{#1}{1}[\LWR@strresult]%
203 \LWR@originname{\LWR@strresult}%
204 \StrChar{#1}{2}[\LWR@strresult]%
205 \LWR@originname{\LWR@strresult}%
206 }
```

Handle the origin key for `\rotatebox`:

```
207 \define@key{krotbox}{origin}{%
208 \renewcommand*{\LWR@rotboxorigin}{#1}%
209 }
```

These keys are ignored:

```
210 \define@key{krotbox}{x}{}
211 \define@key{krotbox}{y}{}
212 \define@key{krotbox}{units}{}

```

`\rotatebox` [*keyval list*] {*angle*} {*text*}

```
213 \LetLtxMacro\LWR@origrotatebox\rotatebox
214
215 \AtBeginDocument{
216 \RenewDocumentCommand{\rotatebox}{O{} m +m}{%
```

Reset the origin to “none-given”:

```
217 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
218 \setkeys{krotbox}{#1}%

```

Select inline-block so that HTML will transform this span:

```
219 \LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
220 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{} }{%
221 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```
222 \LWR@rotstyle{-ms-}{#2} %
223 \LWR@rotstyle{-webkit-}{#2} %
224 \LWR@rotstyle{}{#2} %
225 "}}\LWR@orignewline%
```

Print the text to be rotated:

```
226 \begin{LWR@nestspan}%
227 #3%
```

Close the span:

```
228 \LWR@htmltagc{/span}%
229 \end{LWR@nestspan}%
230 }
231 }% AtBeginDocument
```

`\scalebox` $\{\langle h\text{-scale}\rangle\} [\langle v\text{-scale}\rangle] \{\langle text\rangle\}$

```
232 \LetLtxMacro\LWR@origscalebox\scalebox
233
234 \AtBeginDocument{
235 \RenewDocumentCommand{\scalebox}{m o m}{%
```

Select inline-block so that HTML will transform this span:

```
236 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
237 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
238 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
239 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
240 "{}}%
```

Print the text to be scaled:

```
241 \begin{LWR@nestspan}%
242 #3%
```

Close the span:

```
243 \LWR@htmltagc{/span}%
244 \end{LWR@nestspan}%
245 }
246 }% AtBeginDocument
```

`\reflectbox` $\{\langle text\rangle\}$

```
247 \let\LWR@origreflectbox\reflectbox
248
249 \AtBeginDocument{
250 \renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
251 }
```

`\resizebox` $\{\langle h\text{-length}\rangle\} \{\langle v\text{-length}\rangle\} \{\langle text\rangle\}$

Simply prints its text argument.

```
252 \LetLtxMacro\LWR@origresizebox\resizebox
253
254 \AtBeginDocument{
255 \renewcommand{\resizebox}[3]{#3}
256 }
```

File 71 `lwarp-graphicx.sty`

§ 147 Package **graphicx**

Pkg `graphicx` `graphicx` is emulated.

`graphicx` loads `graphics`, which also loads `lwarp-graphics`, which remembers the original `graphics` definitions for use inside a `lateximage`, and then patches them `\AtBeginDocument` for HTML output.


`lwarp-graphics` handles the syntax of either `graphics` or `graphicx`.

for HTML output: `1 \LWR@ProvidesPackagePass{graphicx}`

File 72 `lwarp-grffile.sty`

§ 148 Package **grffile**

Pkg `grffile` `grffile` is supported as-is. File types known to the browser are displayed, and unknown file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

 **matching PDF and SVG**

`lwarp-grffile` now exists as a placeholder since `grffile` used to be emulated by `lwarp`, and thus older versions of `lwarp-grffile` may exist and should be overwritten by this newer version.

for HTML output: `1 \LWR@ProvidesPackagePass{grffile}`

File 73 `lwarp-hyperref.sty`

§ 149 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 lwarp html version of package 'hyperref' -- discarding options.}
3 \typeout{   Are not using ProvidesPackage, so that other packages}
4 \typeout{   do not attempt to patch lwarp's version of 'hyperref'.}
5 % \ProvidesPackage{lwarp-#1-#2}
6 \DeclareOption*{}
7 \ProcessOptions\relax

8 \newcommand*{\hypersetup}[1]{}
9 \newcommand*{\hyperbaseurl}[1]{}

```

\hyperimage $\{\langle url \rangle\} \{\langle alt \text{ text} \rangle\}$

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@htmltag{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 }

```

\hyperdef $\{\langle 1: \text{category} \rangle\} \{\langle 2: \text{name} \rangle\} \{\langle 3: \text{text} \rangle\}$

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 }

```

\LWR@hyperrefb $\{\langle 1: \text{URL} \rangle\} \{\langle 2: \text{category} \rangle\} \{\langle 3: \text{name} \rangle\} \{\langle 4: \text{text} \rangle\}$

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 }

```

\LWR@hyperrefc $[\langle label \rangle] \{\langle text \rangle\}$

Creates text as an HTML link to the \LaTeX label.

```
32 \NewDocumentCommand{\LWR@hyperrefc}{0{label} +m}{%
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }
```

`\hyperref` $\langle 1: URL \rangle \langle 2: category \rangle \langle 3: name \rangle \langle 4: text \rangle$ — or —
 $[\langle 1: label \rangle] \langle 2: text \rangle$

```
38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode'\_ =12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }
```

`\hypertarget` $\langle name \rangle \langle text \rangle$

Creates an anchor to name with the given text.

```
43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }
```

`\hyperlink` $\langle name \rangle \langle text \rangle$

Creates a link to the anchor created by `hypertarget`, with the given link text.

```
47 \NewDocumentCommand{\hyperlink}{m +m}{%
48 \hyperref[ #1 ]{ #2 }%
49 }
```

`\autoref` * $\langle label \rangle$

For HTML, `\cleveref` is used instead.

```
50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }
```

`\autopageref` $\langle label \rangle$

For HTML, `\cleveref` is used instead.

```
53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }
```

```
\pdfstringdef  {\langle macroname \rangle} {\langle TEXstring \rangle}
56 \newcommand{\pdfstringdef}[2] {}
```

```
\pdfbookmark  [{\langle level \rangle}] {\langle text \rangle} {\langle name \rangle}
57 \newcommand{\pdfbookmark}[3] [] {}
```

```
\currentpdfbookmark  {\langle text \rangle} {\langle name \rangle}
58 \newcommand{\currentpdfbookmark}[2] {}
```

```
\subpdfbookmark  {\langle text \rangle} {\langle name \rangle}
59 \newcommand{\subpdfbookmark}[2] {}
```

```
\belowpdfbookmark  {\langle text \rangle} {\langle name \rangle}
60 \newcommand{\belowpdfbookmark}[2] {}
```

```
\texorpdfstring  {\langle TEXstring \rangle} {\langle PDFstring \rangle}
61 \newcommand{\texorpdfstring}[2] {\#2}
```

```
\hypercalcbp  {\langle dimen \rangle} From hyperref.
62 \def\hypercalcbp#1{%
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
64 }%
```

```
\Acrobatmenu  {\langle menuoption \rangle} {\langle text \rangle}
65 \newcommand{\Acrobatmenu}[2] {}
```

```
\TextField  [{\langle parameters \rangle}] {\langle label \rangle}
66 \newcommand*{\TextField}[2] [] {}
```

```
\CheckBox  [{\langle parameters \rangle}] {\langle label \rangle}
67 \newcommand*{\CheckBox}[2] [] {}
```

```
\ChoiceMenu  [{\langle parameters \rangle}] {\langle label \rangle} {\langle choices \rangle}
68 \newcommand{\ChoiceMenu}[3] [] {}
```

\PushButton [*<parameters>*] {*<label>*}

69 \newcommand*\PushButton[2][]{}

\Submit [*<parameters>*] {*<label>*}

70 \newcommand*\Submit[2][]{}

\Reset [*<parameters>*] {*<label>*}

71 \newcommand*\Reset[2][]{}

\LayoutTextField {*<label>*} {*<field>*}

72 \newcommand*\LayoutTextField[2]{}

\LayoutChoiceField {*<label>*} {*<field>*}

73 \newcommand*\LayoutChoiceField[2]{}

\LayoutCheckField {*<label>*} {*<field>*}

74 \newcommand*\LayoutCheckField[2]{}

\MakeRadioField {*<width>*} {*<height>*}

75 \newcommand*\MakeRadioField[2]{}

\MakeCheckField {*<width>*} {*<height>*}

76 \newcommand*\MakeCheckField[2]{}

\MakeTextField {*<width>*} {*<height>*}

77 \newcommand*\MakeTextField[2]{}

\MakeChoiceField {*<width>*} {*<height>*}

78 \newcommand*\MakeChoiceField[2]{}

\MakeFieldButton {*<text>*}

79 \newcommand*\MakeFieldButton[1]{}

File 74 **lwarp-hyperxmp.sty**

§ 150 Package **hyperxmp**

Pkg hyperxmp Emulated.

for HTML output: Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 75 **lwarp-idxlayout.sty**

§ 151 Package **idxlayout**

Pkg idxlayout Emulated.

for HTML output: Discard all options for lwarp-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}{} }
15
16 \newcommand{\idxlayout}[1]{
17   \newcommand*{\indexfont}{}
18   \newcommand*{\indexjustific}{}
19   \newcommand*{\indexsubsdelim}{}
20   \newcommand*{\indexstheadcase}{}

```

File 76 **lwarp-indentfirst.sty**

§ 152 Package **indentfirst**

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

File 77 **lwarp-inputenc.sty**

§ 153 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 78 **lwarp-keyfloat.sty**

§ 154 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}{0{-1.2ex} m}
15 {% start
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{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 79 **lwarp-layout.sty**

\$ 155 Package **layout**

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}

2 \NewDocumentCommand{\layout}{s}{}
```

File 80 **lwarp-letterspace.sty**

\$ 156 Package **letterspace**

Pkg letterspace letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}

2 \newcommand*\lsstyle{}
3 \newcommand\textls[2] []{}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

File 81 **lwarp-lettrine.sty**

\$ 157 Package **lettrine**

(Based on original code by DANIEL FLIPO.)

Pkg lettrine Emulated.

for HTML output: Discard all options for lwarp-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 82 **lwarp-lips.sty**

§ 158 Package **lips**

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```

1 % \LWR@ProvidesPackageDrop{lips}
2 \PackageInfo{lwarp}{Using the lwarp version of package ‘lips’.%}
3 \ProvidesPackage{lwarp-lips}
4
5 \NewDocumentCommand{\Lips}{-}{\textellipsis}
6
7 \NewDocumentCommand{\BracketedLips}{-}{[\textellipsis]}
8
9 \let\lips\Lips
10 \let\olips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}

```

File 83 **lwarp-listings.sty**

§ 159 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}{lo1}{#1}{#2}}
```

\lst@Init {\backslash-processing}} 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@htmltag{span class="inlineprogramlisting"}}{}%
23 \fi%
24 }

```

`\lst@DeInit` Done at the end of a listing.

```

25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenables line numbers:

```

27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthk@EveryPar%
31 \else%

```

Inline, so create the closing ``:

```

32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%

```

Final listings deinit:

```

34 \LWR@origl@DeInit%
35 }

```

`\lst@MakeCaption` `{\t/b}`

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```

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 % \ifx\lst@label\@empty\else
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 lwarp:

```

55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %                                \addcontentsline{lol}{lstlisting}{\lst@name}
57                                \fi\fi
58                                \fi
59                                \else

```

This would have to be modified for lwarp:

```

60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61                                \addcontentsline{lol}{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 \@tempkipa}%
74 %                                \def\lst@vskip{\nobreak\@vskip\@tempkipa\nobreak}%
75 %                                \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %                                \ifx #1t\allowbreak \fi
77                                \ifx\lst@title\@empty

```

New lwarp code to create a caption:

```

78                                \lst@makecaption\fnnum@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 #1b\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{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\%
114   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
115     \kern\VerbatimHTMLWidth \kern\lst@numbersep
116     \lst@numberstyle{\thelstnumber}}}%
117   }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

118 \end{warpHTML}

```

File 84 **lwarp-longtable.sty**

§ 160 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` * [*horizontalalignment*] [*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{\LTcaption}%
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\LWR@floatend}
13
14 \newenvironment{longtable}[2][{}]{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaption}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaption}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@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 85 **lwarp-lscape.sty**

§ 161 Package **lscape**

Pkg lscape lscape is nullified.

for HTML output: Discard all options for lwarp-lscape.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

File 86 **lwarp-ltcaption.sty**

§ 162 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 lwarp.

longtable* is already defined by lwarp-longtable.

```

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}

```

```
5 \newcommand*{\LTcapmarginfalse}{}

```

File 87 **lwarp-marginfit.sty**

\$ 163 Package **marginfit**

Pkg marginfit Emulated.

for HTML output: Discard all options for lwarp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}

```

File 88 **lwarp-marginfix.sty**

\$ 164 Package **marginfix**

Pkg marginfix Emulated.

for HTML output: Discard all options for lwarp-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][1]{}
10 \newcommand*{\unblockmargin}[1][1]{}
11 \newcommand*{\marginphantom}[2][1]{}

```

File 89 **lwarp-marginnote.sty**

\$ 165 Package **marginnote**

Pkg marginnote Emulated.

for HTML output: Discard all options for lwarp-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 90 **lwarp-mcaption.sty**

§ 166 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 91 **lwarp-mdframed.sty**

§ 167 Package **mdframed**

Pkg mdframed mdframed is loaded with options forced to framemethod=none.

for HTML output: 1 \RequirePackage{xcolor}% for \convertcolorspec
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@stoppars},
10 }
```

`\LWR@mdfprintcolor` `{\mdfcolorkey}`

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 }

\LWR@mdfprintlength {<mdflengthkey>}

Given the mdframed key, print the length.

```
15 \newcommand*{\LWR@mdfprintlength}[1]{%
16 \rndprintlength{\csuse{mdf@#1@length}}
17 }
```

\LWR@mdframedstart 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@htmltagc{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@origspace}%
39 \LetLtxMacro\rule\LWR@origrule%
40 \LetLtxMacro\makebox\LWR@origmakebox%
41 }
```

`\LWR@mdframedend` 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 }
```

`\mdfframedtitleenv` `{\title}`

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@htmltagc{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}{Opt}}
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%

```

Close the span and unnest the paragraph tag restriction:

```

77 \LWR@htmltagc{/span}%
78 \end{LWR@nestspan}%
79 }
80 }

```

```

\LWR@mdfsubtitlecommon {\langle sub -or- subsub \rangle} [\langle options \rangle] {\langle title \rangle}

```

Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubtitle.

Encapsulate the subtitle inside a of class mdframedsubtitle:

```

81 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
82 {% the following empty line is required
83

```

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@htmltagc{/span}% the following empty line is required
110 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
111
112 }
```

`\LWR@mdfsubtitle` [*<options>*] {*<title>*}

```
113 \let\LWR@origmdfsubtitle\mdfsubtitle
114
115 \newcommand*{\LWR@mdfsubtitle}{%
116 \LWR@mdfsubtitlecommon{sub}%
117 }
118 \let\mdfsubtitle\LWR@mdfsubtitle
```

`\LWR@mdfsubsubsubtitle` [*<options>*] {*<title>*}

```
119 \let\LWR@origmdfsubsubsubtitle\mdfsubsubsubtitle
120
121 \newcommand*\LWR@mdfsubsubsubtitle{%
122 \LWR@mdfsubsubsubtitlecommon{subsub}%
123 }
124 \let\mdfsubsubsubtitle\LWR@mdfsubsubsubtitle
```

`\LWR@mdthisenv` Stores the environment of the frame about to be created:

```
125 \newcommand*\LWR@mdthisenv{\mdframed}
```

`\newmdenv` [*<options>*] {*<env-name>*}

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 }
```

`\surroundwithmdframed` [*<options>*] {*<environment>*}

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 }
```

`\mdtheorem` [*<mdframed-options>*] envname [*<numberedlike>*] {*<caption>*} [*<within>*]

Modified from the original to remember the environment.

```
141 \DeclareDocumentCommand{\mdtheorem}{ 0 {} m o m o }%
142 {\ifcsdef{#2}%
143 {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
144 {%
145 \IfNoValueTF {#3}%
146 {%#3 not given -- number relationship
147 \IfNoValueTF {#5}%
148 {%#3+#5 not given
```



```

149 \definecounter{#2}%
150 \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
151 \newenvironment{#2}[1][]{%
152 \refstepcounter{#2}%
153 \ifstrepty{##1}%
154 {\let\@temptitle\relax}%
155 {%
156 \def\@temptitle{\mdf@theoremseparator%
157 \mdf@theoremspace%
158 \mdf@theoremtitlefont%
159 ##1}%
160 \mdf@thm@caption{#2}{\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 \ifstrepty{##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 \ifstrepty{##1}%
179 {\let\@temptitle\relax}%
180 {%
181 \def\@temptitle{\mdf@theoremseparator%
182 \mdf@theoremspace%
183 \mdf@theoremtitlefont%
184 ##1}%
185 \mdf@thm@caption{#2}{\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 \ifstrepty{##1}%
192 {\let\@temptitle\relax}%
193 {%
194 \def\@temptitle{\mdf@theoremseparator%
195 \mdf@theoremspace%
196 \mdf@theoremtitlefont%
197 ##1}%
198 \mdf@thm@caption{#2}{\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         \ifstrepty{##1}%
209             {\let\@temptitle\relax}%
210             {%
211                 \def\@temptitle{\mdf@theoremseparator%
212                     \mdf@theoremspace%
213                     \mdf@theoremtitlefont%
214                     ##1}%
215                 \mdf@thm@caption{#2}{\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         \ifstrepty{##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 }

```

`\newmdtheoremenv` [*mdframed-options*] envname [*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         {%
233             \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
234             \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
235         }%
236     \BeforeBeginEnvironment{#2}{%
237         \renewcommand*\LWR@mdthisenv{\md#2}%
238         \begin{mdframed}[#1]}%
239     \AfterEndEnvironment{#2}{%
240         \end{mdframed}}%
241 }

```

File 92 **lwarp-metalogo.sty**

§ 168 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 93 **lwarp-microtype.sty**

§ 169 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}{}{-}
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 94 **lwarp-midfloat.sty**

§ 170 Package **midfloat**

Pkg midfloat midfloat is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{midfloat}

2 \newenvironment{strip}[1][{}]{-}
3 \newskip\stripsep

```

File 95 **lwarp-moreverb.sty**

§ 171 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@verbatiminput\@verbatiminput
13
14 \renewcommand{\@verbatiminput}[2][]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatiminput[#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=\@one%
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 96 **lwarp-mparhack.sty**

§ 172 Package **mparhack**

Pkg mparhack Emulated.

for HTML output: Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

File 97 **lwarp-multicol.sty**

§ 173 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}

Env multicols * {\langle numcols \rangle} [\langle heading \rangle]

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 \IfValueT{#3}{\begin{BlockClass}{multicolshheading}#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*{\RLmulticolcolumns}{}
11 \newcommand*{\LRmulticolcolumns}{}
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolsep}
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 98 **lwarp-multirow.sty**

§ 174 Package **multirow**

Pkg **multirow** multirow is emulated during HTML output, and used as-is while inside a `lateximage`.

In a lateximage, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 58.18 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}{}%
```

§ 174.1 Multirow

`\multirow` [*⟨vpos⟩*] [*⟨numrows⟩*] [*⟨bigstruts⟩*] [*⟨width⟩*] [*⟨fixup⟩*] [*⟨text⟩*]

```
4 \RenewDocumentCommand{\multirow}{0{c} m o m o +m}%
5 {%
6 \LWR@traceinfo{*** multirow #1 #2 #4}
7 \LWR@maybenewtablerow%
8 \LWR@tabularleftedge%
```

Print the start of a new table data cell:

```
9 \LWR@htmltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
10 \IfValueT{#1}{%
11 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{ }%
12 \ifstrequal{#1}{t}{style="vertical-align:top" }{ }%
13 }%
```

The left/right border, if given:

```
14 \ifdefvoid{\LWR@multirowborder}{ }{%
15 style="border-\LWR@multirowborder: 2px dotted black ; %
16 padding-\LWR@multirowborder: 2px" %
17 }%
```

A class adds the column spec and the rule:

```
18 class="td%
```


Append this column's spec:

```
19 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag. Also add the vertical bar class.

```
20 \LWR@addcmidruletrim%
21 \LWR@addleftmostbartag%
22 \LWR@printbartag{\theLWR@tablecolspos}%
23 "%
24 \LWR@addcmidrulewidth%
25 }%
```

The column's < spec:

```
26 \LWR@gettexpararray{LWR@colbefore-spec}{\theLWR@tablecolspos}%
```

While printing the text, redefine `\` to generate a new line

```
27 \begingroup\LetLtxMacro{\}{\LWR@endoffline}\#6\endgroup%
28 \LWR@stoppars%
29 \global\boolfalse{LWR@intabularmetadata}%
30 \renewcommand{\LWR@multirowborder}{}%
31 \LWR@traceinfo{*** multirow done}%
32 }%
```

§ 174.2 Combined multicolumn and multirow

△ `\multicolumn &`
`\multirow`

lwarp does not support directly combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[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.

△ **skipped cells**

As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines,

△ **empty cells**

`\mcolrowcell` must be used for each cell of each column and each row to be skipped:

```
... & \multicolumnrow{2}{c}[c]{3}[0]{1in}[Opt]{Text} & ...
... & \mcolrowcell & \mcolrowcell & ...
... & \mcolrowcell & \mcolrowcell & ...
```

vposn

Note that recent versions of `multirow` include a new optional `vposn` argument.

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

```
33 \NewDocumentCommand{\multicolumnrow}{m m O{} m O{} m O{} +m}{%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
34 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn/multirow tag:

```
35 \begingroup\LetLtxMacro{\}\{\LWR@endoffline}%
36 \LWR@domulticolumn[#3][#4]{\theLWR@tabhtmlcoltotal}{#2}{#8}%
37 \endgroup%
```

Move to the next \LaTeX column:

```
38 \addtocounter{LWR@tablecolspos}{#1}%
39 \addtocounter{LWR@tablecolspos}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
40 \booltrue{LWR@skipatbang}%
41 }
```

```

42 \appto{\LWR@restoreorigformatting}{%
43 \LetLtxMacro\multirow\LWR@origmultirow%
44 \renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
45 }

```

File 99 **lwarp-nameref.sty**

§ 175 Package **nameref**

Pkg **nameref** **nameref** is emulated by **lwarp**.

for HTML output: Discard all options for **lwarp-nameref**:

```

1 \typeout{Using the lwarp html version of package 'nameref' -- discarding options.}
2 \typeout{    Are not using ProvidesPackage, so that other packages}
3 \typeout{    do not attempt to patch lwarp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax

```

File 100 **lwarp-needspace.sty**

§ 176 Package **needspace**

Pkg **needspace** **needspace** is not used during HTML conversion.

for HTML output: Discard all options for **lwarp-needspace**:

```

1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}

```

File 101 **lwarp-newclue.sty**

§ 177 Package **newclue**

Pkg **newclue** Error if **newclue** is loaded after **lwarp**.

Discard all options for **lwarp-newclue**:

for HTML output: 1 \LWR@ProvidesPackageDrop{newclue}

```
2 \LWR@loadbefore{newclude}
```

File 102 **lwarp-newunicodechar.sty**

§ 178 Package **newunicodechar**

Pkg newunicodechar Error if newunicodechar is loaded after lwarp.

Discard all options for lwarp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}

```
2 \LWR@loadbefore{newunicodechar}
```

File 103 **lwarp-nextpage.sty**

§ 179 Package **nextpage**

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for lwarp-nextpage.

```
1 \LWR@ProvidesPackageDrop{nextpage}

2 \newcommand{\cleartoevenpage}[1] [] {}
3 \newcommand{\movetoevenpage}[1] [] {}
4 \newcommand{\cleartooddpage}[1] [] {}
5 \newcommand{\movetooddpage}[1] [] {}
```

File 104 **lwarp-nonumonpart.sty**

§ 180 Package **nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

File 105 **lwarp-nopageno.sty**

§ 181 Package **nopageno**

Pkg nopageno nopageno is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

File 106 **lwarp-nowidow.sty**

§ 182 Package **nowidow**

Pkg nowidow nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

for HTML output: 1 \LWR@ProvidesPackageDrop{nowidow}

\nowidow [$\langle lines \rangle$]
 \setnowidow [$\langle lines \rangle$]
 2 \newcommand*{\nowidow}[1] [] {}
 3 \newcommand*{\setnowidow}[1] [] {}

\noclub [$\langle lines \rangle$]
 \setnoclub [$\langle lines \rangle$]
 4 \newcommand*{\noclub}[1] [] {}
 5 \newcommand*{\setnoclub}[1] [] {}

File 107 **lwarp-ntheorem.sty**

§ 183 Package **ntheorem**

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg ntheorem ntheorem is patched for use by lwarp.


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 \TeX font settings are ignored.

 **Equation numbering** `ntheorem` has a bug with equation numbering in $\mathcal{A}\mathcal{M}\mathcal{S}$ 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.

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

 **Disabled options** The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because `lwarp` emulated `hyperref`.

for HTML output: Some disabled options:

```

1 \DeclareOption{thref}{}
2
3
4 \newbool{LWR@theoremmarks}
5 \boolfalse{LWR@theoremmarks}
6
7 \DeclareOption{thmmarks}{
8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
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}{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     \theoremstyle{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30   }%
31   {
32     \theoremstyle{#1}
33     \renewcommand{\LWR@newtheoremstyle}{#1}% new
34   }
35 }
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
36
37 \gdef\@xnthm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
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 \@thmcountersep
45       {\noexpand\csname the\theoremnumbering\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}}}% new
52   \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\@ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname the\theoremnumbering\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@thmstyle{\csuse{LWR@thmstyle#1}}}% new
68      \fi
69  }
70
71  \gdef\@othm#1[#2]#3{%
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          \@thm{#1}{#2}{#3}
86        }%
87      \global\@namedef{end#1}{\@endtheorem}
88      \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\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@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/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[\hskip\labelsep \theorem@headerfont
117     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
118   {\item[\hskip\labelsep \theorem@headerfont
119     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
120
121 \renewtheoremstyle{break}%
122   {\item[
123 %   \rlap{\vbox{\hbox{
124     \hskip\labelsep \theorem@headerfont
125     \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 %   }\hbox{\strut}}}]
127   ]}%
128   {\item[
129 %   \rlap{\vbox{\hbox{
130     \hskip\labelsep \theorem@headerfont
131     \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline
132 %   }\hbox{\strut}}}]
133   ]}
134
135 \renewtheoremstyle{change}%
136   {\item[\hskip\labelsep
137     \theorem@headerfont
138     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]}%
139   {\item[\hskip\labelsep
140     \theorem@headerfont
141     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145 %   \rlap{\vbox{\hbox{
146     \hskip\labelsep \theorem@headerfont
147     \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148 %   }\hbox{\strut}}}]
149   ]}%

```

```

150 {\item[
151 % \rlap{\vbox{\hbox{
152     \hskip\labelsep \theorem@headerfont
153     \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
154 % }\hbox{\strut}}}
155 ]}
156
157 \renewtheoremstyle{margin}%
158 {\item[\hskip\labelsep\theorem@headerfont
159     \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
160 ]}%
161 {\item[\hskip\labelsep\theorem@headerfont
162     \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
163 ]}
164
165 \renewtheoremstyle{marginbreak}%
166 {\item[\hskip\labelsep\theorem@headerfont
167     \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
168 ]}%
169 {\item[\hskip\labelsep\theorem@headerfont
170     \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
171 ]}
172
173 \renewtheoremstyle{nonumberplain}%
174 {\item[\theorem@headerfont\hskip\labelsep
175     \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
176 {\item[\theorem@headerfont\hskip \labelsep
177     \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}
178
179 \renewtheoremstyle{nonumberbreak}%
180 {\item[
181 % \rlap{\vbox{\hbox{
182     \hskip\labelsep \theorem@headerfont
183     \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
184 % }\hbox{\strut}}}
185 ]}%
186 {\item[
187 % \rlap{\vbox{\hbox{
188     \hskip\labelsep \theorem@headerfont
189     \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
190 % }\hbox{\strut}}}
191 ]}
192
193 \renewtheoremstyle{empty}%
194 {\item[]}%
195 {\item[\theorem@headerfont \hskip\labelsep\relax
196     \InlineClass{theoremheaderplain}{##3}]}
197
198 \renewtheoremstyle{emptybreak}%
199 {\item[]}%

```

```

200 {\item[\theorem@headerfont \hskip\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@theoremamsthm}{}{%
203 % upright text via CSS
204   \newtheoremstyle{plainupright}{%
205     {\item[\hskip\labelsep \theorem@headerfont
206       \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
207     {\item[\hskip\labelsep \theorem@headerfont
208       \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
209
210 % upright text and small caps header via CSS
211   \newtheoremstyle{nonumberplainuprightsc}{%
212     {\item[\theorem@headerfont\hskip\labelsep
213       \InlineClass{theoremheadersc}{##1\theorem@separator}]]}%
214     {\item[\theorem@headerfont\hskip \labelsep
215       \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]]}

```

The following standard configuration is renewed using the new css:

```

216   \theoremstyle{plainupright}
217   \theorembodyfont{\upshape}
218   \theoremsymbol{\HTMLUnicode{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{\HTMLUnicode{220E}}% UTF-8 end-of-proof
234   \renewtheorem{Proof}{Proof}
235   \renewtheorem{proof}{Proof}
236   \renewtheorem{Beweis}{Beweis}
237   \renewtheorem{beweis}{Beweis}
238   \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
239
240   \theoremsymbol{}
241 }% not amsthm

```

Only if the amsthm option was given:

```

242 \ifbool{LWR@theoremamsthm}{
243
244 \gdef\th@plain{%
245   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
246   \def\@begintheorem##1##2{%
247     \LWR@forcenewpage% new
248     \BlockClass{theorembodyplain}% new
249     \LWR@inctheorem% new
250     \item[\hskip\labelsep
251 % \theorem@headerfont
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[\hskip\labelsep
259 % \theorem@headerfont
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[\hskip\labelsep
270 % \theorem@headerfont
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[\hskip\labelsep
278 % \theorem@headerfont
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[\hskip\labelsep
289 % \theorem@headerfont

```

```

290 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
291 }}%
292 \def\@opargbegintheorem##1##2##3{%
293 \LWR@forcenewpage% new
294 \BlockClass{theorembodydefinition}% new
295 \LWR@inctheorem% new
296 \item[\hskip\labelsep
297 % \theorem@headerfont
298 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
299 ]}}
300
301 \gdef\th@nonumberdefinition{%
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
303 \def\@begintheorem##1##2{%
304 \LWR@forcenewpage% new
305 \BlockClass{theorembodydefinition}% new
306 \LWR@inctheorem% new
307 \item[\hskip\labelsep
308 % \theorem@headerfont
309 \InlineClass{theoremheaderdefinition}{##1.}
310 ]}%
311 \def\@opargbegintheorem##1##2##3{%
312 \LWR@forcenewpage% new
313 \BlockClass{theorembodydefinition}% new
314 \LWR@inctheorem% new
315 \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
318 ]}}
319
320 \gdef\th@remark{%
321 \def\theorem@headerfont{\itshape}\normalfont%
322 \def\@begintheorem##1##2{%
323 \LWR@forcenewpage% new
324 \BlockClass{theorembodyremark}% new
325 \LWR@inctheorem% new
326 \item[\hskip\labelsep
327 % \theorem@headerfont
328 \InlineClass{theoremheaderremark}{##1\ ##2.}
329 ]}%
330 \def\@opargbegintheorem##1##2##3{%
331 \LWR@forcenewpage% new
332 \BlockClass{theorembodyremark}% new
333 \LWR@inctheorem% new
334 \item[\hskip\labelsep
335 % \theorem@headerfont
336 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
337 ]}}
338
339 \gdef\th@nonumberremark{%

```

```

340 \def\theorem@headerfont{\itshape}\normalfont%
341 \def\@begintheorem##1##2{%
342 \LWR@forcenewpage% new
343 \BlockClass{theorembodyremark}% new
344 \LWR@inctheorem% new
345 \item[\hskip\labelsep
346 % \theorem@headerfont
347 \InlineClass{theoremheaderremark}{##1.}
348 ]}%
349 \def\@opargbegintheorem##1##2##3{%
350 \LWR@forcenewpage% new
351 \BlockClass{theorembodyremark}% new
352 \LWR@inctheorem% new
353 \item[\hskip\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##1##2{%
361 \LWR@forcenewpage% new
362 \BlockClass{theorembodyproof}% new
363 \LWR@inctheorem% new
364 \item[\hskip\labelsep
365 % \theorem@headerfont
366 \InlineClass{theoremheaderproof}{##1.}
367 ]}%
368 \def\@opargbegintheorem##1##2##3{%
369 \LWR@forcenewpage% new
370 \BlockClass{theorembodyproof}% new
371 \LWR@inctheorem% new
372 \item[\hskip\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 \fi%
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 {<1: printed type>} {<2: #>} {<3: optional>} {<4: page>}`

```

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@html@elementclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%

```

```

424 \def\thm@list{#1}\thm@processlist
425 \endgroup
426 \LWR@html@elementclassend{nav}{lothm}%
427 }

```

Proof QED symbol:

```

428
429 \newcommand{\qed}{\quad\the\qedsymbol}
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 {<label>}`

```

436 \newcommand*{\thref}[1]{\cref{#1}}

```

File 108 **lwarp-pagenote.sty**

§ 184 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 109 **lwarp-paralist.sty**

§ 185 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 110 **lwarp-parskip.sty**

§ 186 Package **parskip**

Pkg **parskip** parskip is ignored.

for HTML output: Discard all options for lwarp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

File 111 **lwarp-pdfscape.sty**

§ 187 Package **pdfscape**

Pkg **pdfscape** Emulated.

for HTML output: Discard all options for lwarp-pdfscape:

```
1 \LWR@ProvidesPackageDrop{pdfscape}
```

File 112 **lwarp-pdfsync.sty**

§ 188 Package **pdfsync**

Pkg **pdfsync** Emulated.

for HTML output: Discard all options for lwarp-pdfsync:


```
1 \LWR@ProvidesPackageDrop{pdfsync}
```

```
2 \let\pdfsync\relax
3 \let\pdfsyncstart\relax
4 \let\pdfsyncstop\relax
```

File 113 **lwarp-pfnote.sty**

§ 189 Package **pfnote**

Pkg pfnote pfnote is emulated.

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

for HTML output: 1 \LWR@ProvidesPackageDrop{pfnote}

File 114 **lwarp-placeins.sty**

§ 190 Package **placeins**

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

for HTML output: 1 \LWR@ProvidesPackageDrop{placeins}

2 \newcommand*{\FloatBarrier}{}

File 115 **lwarp-prelim2e.sty**

§ 191 Package **prelim2e**

Pkg prelim2e Emulated.

for HTML output: Discard all options for lwarp-prelim2e:

1 \LWR@ProvidesPackageDrop{prelim2e}

2 \newcommand{\PrelimText}{}

3 \newcommand{\PrelimTextStyle}{}

4 \newcommand{\PrelimWords}{}

File 116 **lwarp-quotchap.sty**

§ 192 Package **quotchap**

Pkg quotchap quotchap is emulated.

(Based on original code by KARSTEN TINNEFELD, JAN KLEVER.)

for HTML output: 1 \LWR@ProvidesPackageDrop{quotchap}

```
2 \newcommand{\@quotchap}{}
3 \newlength{\LWR@quotchapwidth}
4
5 \let\@printcites\relax
6
7 \newcommand*{\@iprintcites}{%
```

Place the quotes inside a <div> of class quotchap, of the maximum selected width:

```
8 \uselengthunit{PT}%
9 \begin{BlockClass}[max-width: \rndprintlength{\LWR@quotchapwidth}]{quotchap}
10 %\begin{minipage}{\LWR@quotchapwidth}
11 \@quotchap
12 %\end{minipage}
13 \end{BlockClass}
```

Deactivate the quote printing:

```
14 \global\let\@printcites\relax
15 }
16
17 \NewEnviron{savequote}[1][\linewidth]{%
```

Remember the width, adjusted for HTML, and make the length assignment global, per:

<https://tex.stackexchange.com/questions/300823/why-is-setlength-ineffective-inside-a-tabular-environment>

```
18 \setlength{\LWR@quotchapwidth}{#1*2}%
19 \global\LWR@quotchapwidth=\LWR@quotchapwidth%
```

Remember the body, and activate the quote printing:

```
20 \global\let\@quotchap\BODY
```

```
21 \global\let\@printcites\@iprintcites%
22 }
```

The quotation author is placed inside a <div> of class qauthor:

```
23 \newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

\qsetcnfont is ignored:

```
24 \newcommand{\qsetcnfont}[1]{}
```

File 117 **lwarp-ragged2e.sty**

§ 193 Package **ragged2e**

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-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 118 **lwarp-realscripts.sty**

§ 194 Package **realscripts**

Pkg realscripts realscripts is emulated. See lwarp.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 \textsupersubscript {s O{1} mm} {%
32 \LWR@setrealscriptsalign{#2}%
33 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
34 \textsubscript{#4}\textsuperscript{#3}%
35 }%
36 }

```

File 119 **lwarp-reysize.sty**


§ 195 Package **relsize**

Pkg relsize relsize is patched for use by lwarp.

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

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

While creating SVG math for HTML, the original definitions are 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][1]{%
24 \renewcommand*{\smaller}[1][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 120 **lwarp-romanbar.sty**

§ 196 Package **romanbar**

Pkg romanbar romanbar is patched for use by lwarp.

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 121 **lwarp-romanbarpagenumber.sty**

§ 197 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{romanbarpagenumber}

```

File 122 **lwarp-rotating.sty**

§ 198 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 123 **lwarp-rotfloat.sty**

§ 199 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` $\{\langle 1: type \rangle\} \{\langle 2: placement \rangle\} \{\langle 3: ext \rangle\} [\langle 4: within \rangle]$

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 124 **lwarp-savetrees.sty**

§ 200 Package **savetrees**

Pkg savetrees Emulated.

for HTML output: Discard all options for lwarp-savetrees:

```

1 \LWR@ProvidesPackageDrop{savetrees}

```

File 125 **lwarp-scalefnt.sty**

§ 201 Package **scalefnt**

Pkg scalefnt scalefnt is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{scalefnt}

2 \DeclareRobustCommand\scalefont[1]{}

```

File 126 **lwarp-sectsty.sty**

§ 202 Package **sectsty**

Pkg sectsty sectsty is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{sectsty}

2 \newcommand*{\partfont}          [1] {}
3 \newcommand*{\partnumberfont}    [1] {}
4 \newcommand*{\parttitlefont}     [1] {}
5 \newcommand*{\chapterfont}       [1] {}
6 \newcommand*{\chapternumberfont} [1] {}
7 \newcommand*{\chaptertitlefont}  [1] {}
8 \newcommand*{\sectionfont}       [1] {}
9 \newcommand*{\subsectionfont}    [1] {}
10 \newcommand*{\subsubsectionfont} [1] {}
11 \newcommand*{\paragraphfont}     [1] {}
12 \newcommand*{\subparagraphfont}  [1] {}
13 \newcommand*{\minisecfont}       [1] {}
14 \newcommand*{\allsectionsfont} [1] {}
15 \newcommand{\nohang}{}

```

`\sectionrule` is only to be used in `*font` commands, thus it is ignored.

```

16 \newcommand*{\sectionrule}[5]{}
17
18 \def\ulemheading#1#2{}

```

File 127 **lwarp-setspace.sty**

§ 203 Package **setspace**

Pkg **setspace** `setspace` is not used during HTML conversion.

Discard all options for `lwarp-setspace`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSinglespace}[1]{}
5 \newcommand*{\singlespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}

```

```

17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
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 128 **lwarp-shadow.sty**

§ 204 Package **shadow**

Pkg shadow shadow is emulated.

for HTML output: Discard all options for lwarp-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 129 **lwarp-showidx.sty**

§ 205 Package **showidx**

Pkg showidx showidx is ignored.

for HTML output: Discard all options for lwarp-showidx:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

File 130 **lwarp-showkeys.sty**

§ 206 Package **showkeys**

Pkg showkeys showkeys is ignored.

for HTML output: Discard all options for lwarp-showkeys:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}

```

File 131 **lwarp-sidecap.sty**

§ 207 Package **sidecap**

Pkg sidecap sidecap is nullified.

for HTML output: Discard all options for lwarp-sidecap.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

<http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>
regarding the creation of starred environments with xparse.

```
2 \NewDocumentEnvironment{SCTable}{soo}
3 {\IfValueTF{#3}{\table[#3]}{\table}}
4 {\endtable}

```

```

5
6 \ExplSyntaxOn
7 \cs_new:cpn {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:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}

```

File 132 **lwarp-sidenotes.sty**

§ 208 Package **sidenotes**

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

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  { % unstarred
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 lwarp version of keyfloat:

```

22 \NewDocumentEnvironment{KFLT@sidenotes@marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 \LWR@forcenewpage
26 \LWR@stoppars%
27 \LWR@htmltag{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{KFLT@sidenotes@marginfloat}{figure}}
37   {\end{KFLT@sidenotes@marginfloat}}
38
39 \RenewDocumentEnvironment{marginfigure}{o}
40   {\begin{KFLT@sidenotes@marginfloat}{figure}}
41   {\end{KFLT@sidenotes@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the $\LaTeX 2_{\epsilon}$ `article.cls` source:

```

42 \renewenvironment{figure*}
43   {\@dblfloat{figure}}
44   {\end@dblfloat}
45
46 \renewenvironment{table*}
47   {\@dblfloat{table}}
48   {\end@dblfloat}

```

File 133 **lwarp-siunitx.sty**

§ 209 Package **siunitx**

Pkg siunitx siunitx is patched for use by lwarp.



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{% in case textcomp was not loaded
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 134 **lwarp-soul.sty**

§ 210 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{\LWR@soulcolor}[4]{%
13 \ifcempty{#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*{\capssselect}[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 135 **lwarp-stabular.sty**

§ 211 Package **stabular**

Pkg stabular stabular is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{stabular}

Env stabular [*\vpos*] {*\colspec*}

```
2 \newenvironment{stabular}[2][c]
3 {
4 \renewcommand{\noalign}[1]{\}
5 \begin{tabular}[#1]{#2}
6 }
7 {\end{tabular}}
```

Env stabular {*\width*} [*\vpos*] {*\colspec*}


```
8 \NewDocumentEnvironment{stabular*}{m o m}
9 {
10 \renewcommand{\noalign}[1]{\}
11 \begin{tabular}[#2]{#3}
12 }
13 {\end{tabular}}
```

File 136 **lwarp-subfig.sty**

§ 212 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 {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}
```

The outer minipage allows side-by-side subfloats with `\hfill` between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7   \@tempcnta=\@ne
8   \if@minipage
9     \@tempcnta=\z@
10  \else\ifdim \lastskip=\z@ \else
11    \@tempcnta=\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\@tempcnta
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@captopadj
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}% new
85 \LWR@startpars% new
86 \endgroup\ignorespaces%

```

```

87 }%

\sfs@subcaption {\langle 1 type \rangle} {\langle 2 lof entry \rangle} {\langle 3 caption \rangle}

88 \long\def\sfs@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   \sfs@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 %   \setbox0\vbox{%
104 %     \hb@xt@\the\@tempdima{%
105 %
106 %       \hss
107 %       \parbox[t]{\the\@tempdima}{%
108 %         \caption@make
109 %           {\@nameuse{sub\@captype name}}%
110 %           {\@nameuse{thesub\@captype}}%
111 %           {#3}
112 %       }%
113 %       \hss
114 %     }
115 %   }%
116   \@ifundefined{FBsc@max}%
117   {\box0}%
118   {
119 %   \parbox[t]{\the\@tempdima}{%
120 \LWR@traceinfo{sfs@subcap B1}% new
121   \LWR@htmlblocktag{figcaption}% new
122   \caption@make
123     {\@nameuse{sub\@captype name}}%
124     {\@nameuse{thesub\@captype}}%
125     {#3}
126   \LWR@htmlblocktag{/figcaption}% new
127 \LWR@traceinfo{sfs@subcap B2}% new
128 %   }%
129 %   }%
130 %   {\dimen@\ht0%
131 %     \advance\dimen@\dp0%
132 %     \ifdim\dimen@>\FBsc@max
133 %       \global\FBsc@max\dimen@

```

```

134         \fi
135         \FB@readaux{\let\FBsubcheight\relax}%
136         \ifx\FBsubcheight\relax
137             \def\next{
138 % \parbox[t]{\the\@tempdima}
139             }%
140         \else
141             \def\next{
142 % \parbox[t][\FBsubcheight][t]{\the\@tempdima}
143             }%
144         \fi
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 {\langle caption label\rangle} {\langle caption text\rangle}

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\@capytype\space
209         \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210         \@nameuse{p@sub\@capytype}%
211         \@nameuse{thesub\@capytype}.)}}

```

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@newsfloat\@newsfloat
217
218 \def\@newsfloat[#1]#2{%
219 \LWR@orig@newsfloat[#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 137 **lwarp-supertabular.sty**

§ 213 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{mpsupertabular}{s o m}
51 {\minipage{\linewidth}\supertabular{#3}}
52 {\endsupertabular\endminipage}

```

File 138 **lwarp-syntonly.sty**

§ 214 Package **syntonly**

 Pkg syntonly Emulated.

 for HTML output: Discard all options for lwarp-syntonly:

```
1 \LWR@ProvidesPackageDrop{syntonly}
```

```

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*{\syntaxonly}{\}
6
7 \@onlypreamble\syntaxonly

```

File 139 **lwarp-tables.sty**

§ 215 Package **tables**

Pkg `tables` `tables` is emulated. `\LWR@hline` is used to handle the optional argument when `tables` is loaded.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tables}

2 \newdimen\tablinesep
3 \newdimen\arraylinesep
4 \newdimen\extrarulesep

```

File 140 **lwarp-tabularx.sty**

§ 216 Package **tabularx**

Pkg `tabularx` `tabularx` is emulated by `lwarp`.

for HTML output: Discard all options for `lwarp-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 141 **lwarp-tabulary.sty**

§ 217 Package **tabulary**

Pkg **tabulary** tabulary is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```

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 142 **lwarp-textarea.sty**

§ 218 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@restorettextarea\@empty
```

File 143 **lwarp-textcomp.sty**

§ 219 Package **textcomp**

Pkg textcomp textcomp is patched for use by lwarp.

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

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

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 \let\LWR@origtextlquill\textlquill
7 \let\LWR@origtexttrquill\texttrquill
8 \let\LWR@origtextcircledP\textcircledP
9 \let\LWR@origtexttwelveudash\texttwelveudash
10 \let\LWR@origtextthreequartersemdash\textthreequartersemdash
11 \let\LWR@origtextmho\textmho
12 \let\LWR@origtextnaira\textnaira
13 \let\LWR@origtextpeso\textpeso
14 \let\LWR@origtextrecipe\textrecipe
15 \let\LWR@origtextpertenthousand\textpertenthousand
16 \let\LWR@origtextbaht\textbaht
17 \let\LWR@origtextdiscount\textdiscount
18 \let\LWR@origtextservicemark\textservicemark
19 \LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
20 \LetLtxMacro\LWR@origcapitalogonek\capitalogonek
21 \LetLtxMacro\LWR@origcapitalgrave\capitalgrave
22 \LetLtxMacro\LWR@origcapitalacute\capitalacute
23 \LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex
24 \LetLtxMacro\LWR@origcapitaltilde\capitaltilde
25 \LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
26 \LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
27 \LetLtxMacro\LWR@origcapitalring\capitalring
28 \LetLtxMacro\LWR@origcapitalcaron\capitalcaron
29 \LetLtxMacro\LWR@origcapitalbreve\capitalbreve
30 \LetLtxMacro\LWR@origcapitalmacron\capitalmacron

```

```
31 \LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
32 \LetLtxMacro\LWR@origtextcircled\textcircled
```

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for Lua_{La}T_EX and Xe_{La}T_EX.

```
33 \AtBeginDocument{
34 \ifPDFTeX
35 \renewcommand*{\textdegree}{\HTMLentity{deg}}
36 \renewcommand*{\textcelsius}{\HTMLunicode{2103}}
37 \renewcommand*{\textohm}{\HTMLunicode{2126}}
38 \renewcommand*{\textmu}{\HTMLunicode{00B5}}
39 \renewcommand*{\textlquill}{\HTMLunicode{2045}}
40 \renewcommand*{\textrquill}{\HTMLunicode{2046}}
41 \renewcommand*{\textcircledP}{\HTMLunicode{2117}}
42 \renewcommand*{\texttwelveudash}{\HTMLunicode{2014}}% emdash
43 \renewcommand*{\textthreequartersemdash}{\HTMLunicode{2014}}% emdash
44 \renewcommand*{\textmho}{\HTMLunicode{2127}}
45 \renewcommand*{\textnaira}{\HTMLunicode{20A6}}
46 \renewcommand*{\textpeso}{\HTMLunicode{20B1}}
47 \renewcommand*{\textrecipe}{\HTMLunicode{211E}}
48 \renewcommand*{\textpertenthousand}{\HTMLunicode{2031}}
49 \renewcommand*{\textbaht}{\HTMLunicode{0E3F}}
50 \renewcommand*{\textdiscount}{\}%}
51 \renewcommand*{\textservicemark}{\HTMLunicode{2120}}
52 \else
```

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```
53 \renewcommand*{\textdegree}{°}
54 \renewcommand*{\textcelsius}{°C}
55 \renewcommand*{\textohm}{Ω}
56 \renewcommand*{\textmu}{μ}
57 \renewcommand*{\textlquill}{{}{}}
58 \renewcommand*{\textrquill}{{}{}}
59 \renewcommand*{\textcircledP}{Ⓟ}
60 \renewcommand*{\texttwelveudash}{--}% emdash
61 \renewcommand*{\textthreequartersemdash}{--}% emdash
62 \renewcommand*{\textmho}{℧}
63 \renewcommand*{\textnaira}{₦}
64 \renewcommand*{\textpeso}{₱}
65 \renewcommand*{\textrecipe}{℞}
66 \renewcommand*{\textpertenthousand}{‰}
67 \renewcommand*{\textbaht}{฿}
68 \renewcommand*{\textdiscount}{\}%}
69 \renewcommand*{\textservicemark}{®}
70 \fi
```

For HTML, Unicode diacritical marks are used:

```

71 \renewcommand*{\capitalcedilla}[1]{\#1\HTMLUnicode{0327}}
72 \renewcommand*{\capitalogonek}[1]{\#1\HTMLUnicode{0328}}
73 \renewcommand*{\capitalgrave}[1]{\#1\HTMLUnicode{0300}}
74 \renewcommand*{\capitalacute}[1]{\#1\HTMLUnicode{0301}}
75 \renewcommand*{\capitalcircumflex}[1]{\#1\HTMLUnicode{0302}}
76 \renewcommand*{\capitaltilde}[1]{\#1\HTMLUnicode{0303}}
77 \renewcommand*{\capitaldieresis}[1]{\#1\HTMLUnicode{0308}}
78 \renewcommand*{\capitalhungarumlaut}[1]{\#1\HTMLUnicode{30B}}
79 \renewcommand*{\capitalring}[1]{\#1\HTMLUnicode{30A}}
80 \renewcommand*{\capitalcaron}[1]{\#1\HTMLUnicode{30C}}
81 \renewcommand*{\capitalbreve}[1]{\#1\HTMLUnicode{306}}
82 \renewcommand*{\capitalmacron}[1]{\#1\HTMLUnicode{304}}
83 \renewcommand*{\capitaldotaccent}[1]{\#1\HTMLUnicode{307}}

```

`\textcircled` becomes a span with a rounded border:

```

84 \renewcommand*{\textcircled}[1]{%
85 \InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{\#1}%
86 }
87 }% AtBeginDocument

```

When a `lateximage` is begun:

```

88 \appto{\LWR@restoreorigformatting}{%
89 \let\textdegree\LWR@origtextdegree%
90 \let\textcelsius\LWR@origtextcelsius%
91 \let\textohm\LWR@origtextohm%
92 \let\textmu\LWR@origtextmu%
93 \let\textlquill\LWR@origtextlquill%
94 \let\textrqquill\LWR@origtextrquill%
95 \let\textcircledP\LWR@origtextcircledP%
96 \let\texttwelveudash\LWR@origtexttwelveudash%
97 \let\textthreequartersemdash\LWR@origtextthreequartersemdash%
98 \let\textmho\LWR@origtextmho%
99 \let\textnaira\LWR@origtextnaira%
100 \let\textpeso\LWR@origtextpeso%
101 \let\textrecipe\LWR@origtextrecipe%
102 \let\textpertenthousand\LWR@origtextpertenthousand%
103 \let\textbaht\LWR@origtextbaht%
104 \let\textdiscount\LWR@origtextdiscount%
105 \let\textservicemark\LWR@origtextservicemark%
106 \LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
107 \LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
108 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
109 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
110 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
111 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%

```

```

112 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
113 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
114 \LetLtxMacro\capitalring\LWR@origcapitalring%
115 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
116 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
117 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
118 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
119 \LetLtxMacro\textcircled\LWR@origtextcircled%
120 }

```

File 144 **lwarp-textpos.sty**

§ 220 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()}{ }{}
3 \NewDocumentEnvironment{textblock*}{m o r()}{ }{}
4 \newcommand*{\TPGrid}[3][ ]{}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*{\textblockcolour}[1]{}
7 \newcommand*{\textblockrulecolour}[1]{}
8 \newcommand*{\textblockcolor}[1]{}
9 \newcommand*{\textblockrulecolor}[1]{}
10 \newcommand*{\tekstblokkulur}[1]{}
11 \newcommand*{\tekstblokrulekulur}[1]{}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}[1]{}
16 \newcommand*{\showtextsize}{}
17 \newcommand{\textblockorigin}[2]{}

```

File 145 **lwarp-theorem.sty**

§ 221 Package **theorem**

(Based on original code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by lwarp.

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\xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\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\@ifdefinable\csname #1\endcsname
35 {
36 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37 \@definecounter{#1}%
38 \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39 \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40 \expandafter{\csname th@\the \theoremstyle \expandafter
41 \endcsname \the\theorembodyfont \@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\@ifdefinable\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 \@tempa
56 \expandafter{\csname th@\the \theoremstyle \expandafter
57 \endcsname \the\theorembodyfont \@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[\hskip\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[\hskip \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[\hskip \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[\hskip \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[\hskip \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[\hskip \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[\hskip \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[\hskip\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[\hskip\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[\hskip\labelsep
144       \InlineClass{theoremheader}{##2 \quad ##1}
145     ]}%
146 \def\@opargbegintheorem##1##2##3{%
147 \LWR@forcenewpage% new
148   \BlockClass{theorembody\LWR@thisthmstyle}% new
149   \item[\hskip\labelsep
150     \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
151   ]}
152 }

```

Patched for CSS:

```

153 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

File 146 **lwarp-threeparttable.sty**

§ 222 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}

```

\LWR@printtablenote  {\langle text \rangle}
Prints the table note item header inside a css class of tnoteitemheader.
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}

Env  threeparttable  [\langle alignment \rangle] To emulate threeparttable:
3 \newenvironment*{threeparttable}[1][b]{\{}{\}}

Env  tablenotes      [\langle options \rangle]
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 }

\tnote  {\langle text \rangle}
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}

```

File 147 **lwarp-tikz.sty**

§ 223 Package **tikz**

Pkg tikz tikz is supported.

Accept all options for lwarp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

catcodes lwarp 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, lwarp 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}% Test for Tikz version v3.0.0
6 {\usetikzlibrary{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}% Test for Tikz version v3.0.0
11 {}%
12 {\catcode'\$=3} % dollar sign is math shift
13 }
14
15 \AfterEndEnvironment{tikzpicture}{%
16 \endlateximage%
17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
18 {}%
19 {\catcode'\$=\active}%
20 }

21 \end{warpHTML}
```

File 148 **lwarp-titles.sty**

§ 224 Package **titles**

Pkg `titles` `titles` is loaded and used by `lwarp` during HTML output. All user options and macros are ignored and disabled.

Discard all options for `lwarp-titles`:

```
for HTML output: 1 \LWR@ProvidesPackageDrop{titles}
```

`\pagestyle` and `\thispagestyle` are already disabled in the `lwarp` code.

```
\newpagestyle {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
```

```

\renewpagestyle  {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
                2 \NewDocumentCommand{\newpagestyle}{m o m}{\}
                3 \NewDocumentCommand{\renewpagestyle}{m o m}{\}

\sethead         [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}
\setfoot         [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}
                4 \NewDocumentCommand{\sethead}{o o o m m m}{\}
                5 \NewDocumentCommand{\setfoot}{o o o m m m}{\}

\settitledmarks  * {\langle names \rangle}
                6 \NewDocumentCommand{\settitledmarks}{s m}{\}

\headrule
\footrule
                7 \newcommand*{\headrule}{\}
                8 \newcommand*{\footrule}{\}

\setheadrule     {\langle length \rangle}
\setfootrule     {\langle length \rangle}
                9 \newcommand*{\setheadrule}[1]{\}
               10 \newcommand*{\setfootrule}[1]{\}

\makeheadrule
\makefootrule
               11 \newcommand*{\makeheadrule}{\}
               12 \newcommand*{\makefootrule}{\}

\setmarkboth     {\langle code \rangle}
               13 \newcommand{\setmarkboth}[1]{\}

\widenhead
               14 \NewDocumentCommand{\widenhead}{s o o m m}{\}

\bottommarks
\topmarks
\firstmarks
\nextmarks

```

\outertitlemarks

\innertitlemarks

```
15 \newcommand*{\bottitlemarks}{}
16 \newcommand*{\toptitlemarks}{}
17 \newcommand*{\firsttitlemarks}{}
18 \newcommand*{\nexttoptitlemarks}{}
19 \newcommand*{\outertitlemarks}{}
20 \newcommand*{\innertitlemarks}{}

```

\newtitlemark * {\langle name \rangle}

```
21 \NewDocumentCommand{\newtitlemark}{s m}{}

```

\pretitlemark * {\langle section \rangle} {\langle text \rangle}

```
22 \NewDocumentCommand{\pretitlemark}{s m m}{}

```

\ifsamemark {\langle group \rangle} {\langle command \rangle} {\langle true \rangle} {\langle false \rangle}

```
23 \newcommand{\ifsamemark}[4]{}

```

\setfloathead * [\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]

\setfloatfoot * [\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]

```
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}

```

\nextfloathead * [\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]

\nextfloatfoot * [\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]

```
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}

```

\newmarkset {\langle markset \rangle}

```
28 \newcommand{\newmarkset}[1]{}

```

\newextramark * {\langle markset \rangle} {\langle macro-name \rangle}

```
29 \NewDocumentCommand{\newextramarkset}{s m m}{}

```

\botextramarks {\langle markset \rangle}

\topextramarks {\langle markset \rangle}

\firstextramarks {\langle markset \rangle}

```

\nextextramarks   {\langle markset \rangle}
\outerextramarks  {\langle markset \rangle}
\innerextramarks  {\langle markset \rangle}

30 \newcommand{\botextramarks}[1]{}
31 \newcommand{\topextramarks}[1]{}
32 \newcommand{\firstextramarks}[1]{}
33 \newcommand{\nextttopextramarks}[1]{}
34 \newcommand{\outerextramarks}[1]{}
35 \newcommand{\innerextramarks}[1]{}

```

File 149 **lwarp-titleref.sty**

§ 225 Package **titleref**

Pkg titleref titleref is superceded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

File 150 **lwarp-titlesec.sty**

§ 226 Package **titlesec**

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

```

\titleref    {\langle label-format \rangle}

2 \newcommand*{\titleref}[1]{}

```

```

\titleref*    {\langle command \rangle} {\langle format \rangle}

```

```

\titleref    {\langle command \rangle} [\langle shape \rangle] {\langle format \rangle} {\langle label \rangle} {\langle sep \rangle} {\langle before \rangle} [\langle after \rangle]

3 \newcommand\titleref{%
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}{-}

```

```

\chaptertitlename
    8 \@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
    9 \newcommand\chaptertitlename{\@chapapp}

\titlespacing * {<command>} {<left>} {<before>} {<after>} [<right>]
    10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

\filright
\filcenter
\filleft
\fillast
\filinner
\filouter
    11 \newcommand*\filright{}
    12 \newcommand*\filcenter{}
    13 \newcommand*\filleft{}
    14 \newcommand*\fillast{}
    15 \newcommand*\filinner{}
    16 \newcommand*\filouter{}

\wordsep
    17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
    18 \fontdimen\thr@@\font \@minus \fontdimen4\font}

\titeline * [<align>] {<material>}
    19 \NewDocumentCommand{\titeline}{s o m}{}

\titlerule [<height>]
    20 \providecommand\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
    21 \newcommand*\ttl@rule[1] [] {}
    22 \newcommand*\ttl@row[2] [] {}

\iftitlemeasuring {<true>} {<false>}
\assignpagestyle {<command>} {<pagestyle>}
\titleclass {<name>} [<startlevel>] {<class>} [<cmd>]
    23 \newcommand\iftitlemeasuring[2] {#2}
    24 \newcommand\assignpagestyle[2] {#2}
    25 \NewDocumentCommand\titleclass{m o m o}

```

File 151 **lwarp-titletoc.sty**

§ 227 Package **titletoc**

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output: 1 \LWR@ProvidesPackageDrop{titletoc}

\dottedcontents {\<section>} [\<left>] {\<above>} {\<label>} {\<leader>}
 2 \NewDocumentCommand{\dottedcontents}{m o m m m}{}

\titlecontents * {\<section>} [\<left>] {\<above>} {\<numbered>} {\<numberless>} {\<filler>} [\<below
 or begin>] [\<separator>] [\<end>]
 3 \newcommand{\titlecontents}{\@ifstar{\ttl@tcstar}{\ttl@tcnostar}}
 4 \NewDocumentCommand{\ttl@tcstar}{m o m m m m o o o}{}
 5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m m o}{}
 6 \newcommand{\titlecontents}{}

\contentsmargin [\<correction>] {\<right>}
 6 \newcommand{\contentsmargin}[2] [] {}

\thecontentslabel
 \thecontentspage
 7 \newcommand*{\thecontentslabel}{\thecontentslabel}
 8 \newcommand*{\thecontentspage}{\thecontentspage}

\contentslabel [\<format>] {\<space>}
 \contentspage [\<format>]
 9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}
 10 \newcommand{\contentspage}[1] [] {\thecontentspage}

\contentspush {\<text>}
 11 \newcommand{\contentspush}[1] {}

\contentsuse {\<name>} {\<text>}

```

12 \newcommand{\contentsuse}[2]{}

\startcontents  [<name>]
\stopcontents   [<name>]
\resumecontents [<name>]

13 \newcommand*{\startcontents}[1] [] {}
14 \newcommand*{\stopcontents}[1] [] {}
15 \newcommand*{\resumecontents}[1] [] {}

\printcontents  [<name>] {\<prefix>} {\<start>} {\<code>}

16 \newcommand{\printcontents}[4] [] {}

\startlist      [<name>] {\<list>}
\stoplist       [<name>] {\<list>}
\resumelist     [<name>] {\<list>}

17 \newcommand{\startlist}[2] [] {}
18 \newcommand{\stoplist}[2] [] {}
19 \newcommand{\resumelist}[2] [] {}

\printlist      [<name>] {\<list>} {\<prefix>} {\<code>}


20 \newcommand{\printlist}[4] [] {}

```

File 152 **lwarp-titling.sty**

§ 228 Package **titling**

Pkg titling

package support lwarp supports the native \LaTeX titling commands, and also supports the packages
 **load order** 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.

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 ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \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 \@bsmarkseries
```

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 \@bscontmark
48 }
```

`\@maketitle` Typesets the title, etc. Patched for HTML.

```
49 \DeclareDocumentCommand{\@maketitle}{*}{%
50   \maketitlehooka
51   {
52     \LWR@stoppars\LWR@htmltag{h1}
53     \@bsprefixtitle \@title \@bsprefixtitle
54     \LWR@htmltag{/h1}\LWR@startpars
55   }
56   \maketitlehookb
57   {
58     \begin{BlockClass}{author}
59     \renewcommand{\and}{\and}
60     \end{BlockClass}
61     \begin{BlockClass}{oneauthor}
62     }
63     \begin{BlockClass}{oneauthor}
64     \@bsprefixauthor \@author \@bsprefixauthor
65     \end{BlockClass}
66     \end{BlockClass}
67   }
68   \maketitlehookc
69   {
```

```

70      \begin{BlockClass}{titledate}
71      \@bspredate \@date \@bspostdate
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 \@bsmarkseries
```

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` `{\langle series \rangle}`

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 153 **lwarp-tocbibind.sty**

§ 229 Package **tocbibind**

Pkg tocbibind tocbibind is patched for use by lwarp.

Opt IndexLanguage The lwarp 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.

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

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` Use the `tocbibind numindex` option to generate a numbered index. Without this option, the index heading has no number.

numbered index section

for HTML output:

```

1 \let\simplechapterdelim\relax
2
3 \LWR@ProvidesPackagePass{tocbibind}

4 \renewenvironment{theindex}%
5   {%
6     \if@bibchapter
7       \if@donumindex
8         \chapter{\indexname}
9       \else
10        \if@dotocind
11          \chapter*{\indexname}
12          \addcontentsline{toc}{chapter}{\indexname}
13        \else
14          \chapter*{\indexname}
15        \fi
16      \fi
17    \else
18      \if@donumindex
19        \section{\indexname}
20      \else
21        \if@dotocind
22          \section*{\indexname}
23          \addcontentsline{toc}{\@tocextra}{\indexname}
24        \else
25          \section*{\indexname}
26        \fi
27      \fi
28    \fi
29 \let\item\LWR@indexitem%
30 \let\subitem\LWR@indexsubitem%
31 \let\subsubitem\LWR@indexsubsubitem%

```



```
32 }
```


The following code is shared by anonchap.

```
33 \renewcommand{\simplechapter}[1][\@empty]{%
34   \def\@chapcntformat##1{%
35     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
36   }%
37 }
38
39 \renewcommand{\restorechapter}{%
40 \let\@chapcntformat\@secCNTformat%
41 }
```

File 154 **lwarp-tocloft.sty**

§ 230 Package **tocloft**

Pkg tocloft tocloft is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

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

Discard all options for lwarp-tocloft:

for HTML output: 1 \LWR@ProvidesPackageDrop{tocloft}

```
\tocloftpagestyle {\style}
2 \newcommand{\tocloftpagestyle}[1]{}

```

```
\cftmarktoc
\cfttoctitlefont
\cftaftertoctitle

3 \newcommand*{\cftmarktoc}{}
4 \newcommand*{\cfttoctitlefont}{}
5 \newcommand*{\cftaftertoctitle}{}

6 \newlength{\cftbeforetoctitleskip}
7 \newlength{\cftaftertoctitleskip}

```

```

\cftmarklof
\cftlofttitlefont
\cftafterlofttitle

8 \newcommand*{\cftmarklof}{}
9 \newcommand*{\cftlofttitlefont}{}
10 \newcommand*{\cftafterlofttitle}{}

11 \newlength{\cftbeforelofttitleskip}
12 \newlength{\cftafterlofttitleskip}

\cftmarklot
\cftlottitlefont
\cftafterlottitle

13 \newcommand*{\cftmarklot}{}
14 \newcommand*{\cftlottitlefont}{}
15 \newcommand*{\cftafterlottitle}{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

\cftdot
\cftdotsep
\cftnodots

18 \newcommand*{\cftdot}{.}
19 \providecommand*{\cftdotsep}{1}
20 \newcommand*{\cftnodots}{5000}

\cftdotfill {\langle sep \rangle}

21 \providecommand{\cftdotfill}[1]{}

\cftsetpnumwidth {\langle length \rangle}
\cftsetrmarg {\langle length \rangle}

22 \newcommand*{\cftsetpnumwidth}[1]{}
23 \newcommand*{\cftsetrmarg}[1]{}

\cftpnumalign {\langle alignment \rangle}

24 \newcommand*{\cftpnumalign}[1]{}

25 \newlength{\cftparskip}

```

```
26 \newlength{\cftbeforepartskip}
27 \newlength{\cftpartindent}
28 \newlength{\cftpartnumwidth}
29 \newcommand*{\cftpartfont}{}
30 \newcommand*{\cftpartpresnum}{}
31 \newcommand*{\cftpartaftersnum}{}
32 \newcommand*{\cftpartaftersnumb}{}
33 \newcommand*{\cftpartleader}{}
34 \newcommand*{\cftpartdotsep}{1}
35 \newcommand*{\cftpartpagefont}{}
36 \newcommand*{\cftpartafterpnum}{}

37 \newlength{\cftbeforechapskip}
38 \newlength{\cftchapindent}
39 \newlength{\cftchapnumwidth}
40 \newcommand*{\cftchapfont}{}
41 \newcommand*{\cftchappresnum}{}
42 \newcommand*{\cftchapaftersnum}{}
43 \newcommand*{\cftchapaftersnumb}{}
44 \newcommand*{\cftchapleader}{}
45 \newcommand*{\cftchapdotsep}{1}
46 \newcommand*{\cftchappagefont}{}
47 \newcommand*{\cftchapafterpnum}{}

48 \newlength{\cftbeforeseccskip}
49 \newlength{\cftsecindent}
50 \newlength{\cftsecnumwidth}
51 \newcommand*{\cftsecfont}{}
52 \newcommand*{\cftsecpresnum}{}
53 \newcommand*{\cftsecaftersnum}{}
54 \newcommand*{\cftsecaftersnumb}{}
55 \newcommand*{\cftsecleader}{}
56 \newcommand*{\cftsecdotsep}{1}
57 \newcommand*{\cftsecpagefont}{}
58 \newcommand*{\cftsecafterpnum}{}

59 \newlength{\cftbeforesubsecskip}
60 \newlength{\cftsubsecindent}
61 \newlength{\cftsubsecnumwidth}
62 \newcommand*{\cftsubsecfont}{}
63 \newcommand*{\cftsubsecpresnum}{}
64 \newcommand*{\cftsubsecaftersnum}{}
65 \newcommand*{\cftsubsecaftersnumb}{}
66 \newcommand*{\cftsubsecleader}{}
67 \newcommand*{\cftsubsecdotsep}{1}
68 \newcommand*{\cftsubsecpagefont}{}
69 \newcommand*{\cftsubsecafterpnum}{}

70 \newlength{\cftbeforesubsubsecskip}
```

```
71 \newlength{\cftsubsubsecindent}  
72 \newlength{\cftsubsubsecnumwidth}  
73 \newcommand*{\cftsubsubsecfont}{}  
74 \newcommand*{\cftsubsubsecpresnum}{}  
75 \newcommand*{\cftsubsubsecaftersnum}{}  
76 \newcommand*{\cftsubsubsecaftersnumb}{}  
77 \newcommand*{\cftsubsubsecleader}{}  
78 \newcommand*{\cftsubsubsecdotsep}{1}  
79 \newcommand*{\cftsubsubsecpagefont}{}  
80 \newcommand*{\cftsubsubsecafterpnum}{}  

```

```
81 \newlength{\cftbeforeparaskip}  
82 \newlength{\cftparaindent}  
83 \newlength{\cftparanumwidth}  
84 \newcommand*{\cftparafont}{}  
85 \newcommand*{\cftparapresnum}{}  
86 \newcommand*{\cftparaaftersnum}{}  
87 \newcommand*{\cftparaaftersnumb}{}  
88 \newcommand*{\cftparaleader}{}  
89 \newcommand*{\cftparadotsep}{1}  
90 \newcommand*{\cftparapagefont}{}  
91 \newcommand*{\cftparaafterpnum}{}  

```

```
92 \newlength{\cftbeforesubparaskip}  
93 \newlength{\cftsubparaindent}  
94 \newlength{\cftsubparanumwidth}  
95 \newcommand*{\cftsubparafont}{}  
96 \newcommand*{\cftsubparapresnum}{}  
97 \newcommand*{\cftsubparaaftersnum}{}  
98 \newcommand*{\cftsubparaaftersnumb}{}  
99 \newcommand*{\cftsubparaleader}{}  
100 \newcommand*{\cftsubparadotsep}{1}  
101 \newcommand*{\cftsubparapagefont}{}  
102 \newcommand*{\cftsubparaafterpnum}{}  

```

```
103 \newlength{\cftbeforefigskip}  
104 \newlength{\cftfigindent}  
105 \newlength{\cftfignumwidth}  
106 \newcommand*{\cftfigfont}{}  
107 \newcommand*{\cftfigpresnum}{}  
108 \newcommand*{\cftfigaftersnum}{}  
109 \newcommand*{\cftfigaftersnumb}{}  
110 \newcommand*{\cftfigleader}{}  
111 \newcommand*{\cftfigdotsep}{1}  
112 \newcommand*{\cftfigpagefont}{}  
113 \newcommand*{\cftfigafterpnum}{}  

```

```
114 \newlength{\cftbeforesubfigskip}  
115 \newlength{\cftsubfigindent}
```

```

116 \newlength{\cftsubfignumwidth}
117 \newcommand*{\cftsubfigfont}{}
118 \newcommand*{\cftsubfigpresnum}{}
119 \newcommand*{\cftsubfigaftersnum}{}
120 \newcommand*{\cftsubfigaftersnumb}{}
121 \newcommand*{\cftsubfigleader}{}
122 \newcommand*{\cftsubfigdotsep}{1}
123 \newcommand*{\cftsubfigpagefont}{}
124 \newcommand*{\cftsubfigafterpnum}{}

125 \newlength{\cftbeforetabskip}
126 \newlength{\cfttabindent}
127 \newlength{\cfttabnumwidth}
128 \newcommand*{\cfttabfont}{}
129 \newcommand*{\cfttabpresnum}{}
130 \newcommand*{\cfttabaftersnum}{}
131 \newcommand*{\cfttabaftersnumb}{}
132 \newcommand*{\cfttableader}{}
133 \newcommand*{\cfttabdotsep}{1}
134 \newcommand*{\cfttabpagefont}{}
135 \newcommand*{\cfttabafterpnum}{}

136 \newlength{\cftbeforesubtabskip}
137 \newlength{\cftsubtabindent}
138 \newlength{\cftsubtabnumwidth}
139 \newcommand*{\cftsubtabfont}{}
140 \newcommand*{\cftsubtabpresnum}{}
141 \newcommand*{\cftsubtabaftersnum}{}
142 \newcommand*{\cftsubtabaftersnumb}{}
143 \newcommand*{\cftsubtableader}{}
144 \newcommand*{\cftsubtabdotsep}{1}
145 \newcommand*{\cftsubtabpagefont}{}
146 \newcommand*{\cftsubtabafterpnum}{}

147 \newcommand{\cftsetindents}[3]{}

148 \newcommand{\pagenumbersoff}[1]{}
149 \newcommand{\pagenumberson}[1]{}

```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]
 Emulated through the `\newfloat` mechanism.

```

150 \NewDocumentCommand{\newlistof}{o m m m}
151 {%
152   \IfValueTF{#1}
153   {\newfloat{#2}{tbp}{#3}[#1]}
154   {\newfloat{#2}{tbp}{#3}}
155   \@namedef{listof#2}{\listof{#2}{#4}}

```

```

156 \@namedef{#2depth}{1}
157 \expandafter\newlength\csuse{cftbefore#2skip}
158 \expandafter\newlength\csuse{cft#2indent}
159 \expandafter\newlength\csuse{cft#2numwidth}
160 \@namedef{cft#2font}{}
161 \@namedef{cft#2presnum}{}
162 \@namedef{cft#2aftersnum}{}
163 \@namedef{cft#2aftersnumb}{}
164 \@namedef{cft#2leader}{}
165 \@namedef{cft#2dotsep}{1}
166 \@namedef{cft#2pagefont}{}
167 \@namedef{cft#2afterpnum}{}
168 }

```

\cftchapterprecis from tocloft:

```

169 \newcommand{\cftchapterprecis}[1]{%
170   \cftchapterprecishere{#1}
171   \cftchapterprecistoc{#1}}
172 \newcommand{\cftchapterprecishere}[1]{%
173   \begin{quote}\textit{#1}\end{quote}}
174 \newcommand{\cftchapterprecistoc}[1]{
175   \addtocontents{toc}{%
176     {
177       \protect\begin{quote}#1\protect\end{quote}}
178   }
179 }

```

File 155 **lwarp-transparent.sty**

§ 231 Package **transparent**

(Based on original code by HEIKO OBERDIEK.)

Pkg transparent Emulated. \texttransparent works for inline objects. \transparent only works for \includegraphics.

 **Not Xe_{La}TeX!** Note that transparent does not work with Xe_{La}TeX.

for HTML output: Discard all options for lwarp-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 156 **lwarp-trivfloat.sty**

§ 232 Package **trivfloat**

Pkg trivfloat trivfloat is forced to use the built-in lwarp emulation for floats.

Discard all options for lwarp-trivfloat. This tells trivfloat not to use floatrow or memoir.

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 lwarp.

for HTML output: 5 \begin{warpHTML}

```

6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}

```

```

7 \end{warpHTML}

```

§ 232.1 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 157 **lwarp-typearea.sty**

§ 233 Package **typearea**

Pkg `typearea` `typearea` is emulated.

for HTML output: `1 \LWR@ProvidesPackageDrop{typearea}`

```
2 \newcommand*\typearea[2] [\z@]{}
3 \newcommand*\recalc{typearea}{}
4 \@ifundefined{footheight}{\newlength{footheight}}{}
5 \newcommand*\areaset[3] [\z@]{}

```

File 158 **lwarp-ulem.sty**

§ 234 Package **ulem**

(Based on original code by DONALD ARSENEAU.)

Pkg ulem Emulated.

for HTML output:

Original lwarp definitions:

```
1 \LetLtxMacro\LWR@ulemorigemph\emph
2 \LetLtxMacro\LWR@ulemorigtextbf\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 159 **lwarp-upref.sty**

\$ 235 Package **upref**

 Pkg upref Ignored.

for HTML output: Discard all options for lwarp-upref:

```

1 \LWR@ProvidesPackageDrop{upref}

```

File 160 **lwarp-verse.sty**

\$ 236 Package **verse**

(Based on original code by PETER WILSON.)

 Pkg verse verse is supported and patched by lwarp.

for HTML output: Pass all options for lwarp-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. `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 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.

`Len \leftmargini`
`Len \TMLvleftskip`
`Len \TMLleftmargini`

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 `\flagverse` command. Do not allow them to go into the left margin, which would cause `pdfcrop` to crop the entire page further to the left:

`Pkg memoir`
`\flagverse`
`Len \leftskip`

```
6 \ifdef{\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{\@vstypeptitle}{m}{%
20   \vspace{\beforepoemtitleskip}%
21   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22   \vspace{\afterpoemtitleskip}%
23 }
24 }{}
25
26 }% AfterEndPreamble

```

File 161 **lwarp-wallpaper.sty**

§ 237 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 162 **lwarp-wrapfig.sty**

§ 238 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@htmltag{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 163 **lwarp-xcolor.sty**

§ 239 Package **xcolor**

Pkg xcolor xcolor is supported by lwarp.

\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.

§ 239.1 **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 68.1. 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 340.

`\color`:

Print: Used as-is.

HTML: Ignored by `pdftotext`, and will not appear.

HTML `lateximage`: Colors will appear in a `lateximage`.

`\textcolor`:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 570.

HTML `lateximage`: Remembers and reuses the print version.

\pagecolor:

Print: Used as-is.

HTML: Ignored.

HTML lateximage: Colors will be picked up in a lateximage.

\nopagecolor:

Print: Used as-is.

HTML: Ignored.

HTML lateximage: Colors will be picked up in a lateximage.

\colorbox:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 570.

HTML lateximage: Remembers and reuses the print version.

\colorboxBlock:

Print: Becomes \colorbox.

HTML: Newly defined by lwarp-xcolor to use a <div>, page 571.

HTML lateximage: Remembers and reuses the print version \colorbox.

\fcolorbox:

Print: Modified to allow a background of none.

\LWRprint@fcolorbox at section 68.1

HTML: Redefined by lwarp-xcolor, page 571.

HTML lateximage: Remembers and reuses the print version.

\fcolorboxBlock:

Print: Becomes \fcolorbox. Section 68.1

HTML: Newly defined by lwarp-xcolor to use a <div>, page 572.

HTML lateximage: Remembers and reuses the print version \fcolorbox.

fcolorminipage:

Print: Newly defined in the lwarp core.

LWRprint@fcolorminipage at section 68.1

HTML: Newly defined by lwarp-xcolor, page 572.

HTML lateximage: Uses the print version.

\boxframe:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 573.

HTML lateximage: Remembers and reuses the print version.

§ 239.2 **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}{}%
```

`\LWR@colorstyle` $\{\langle 1: styletext \rangle\} \{\langle 2: model \rangle\} \{\langle 3: color \rangle\}$

For a color style, prints the color converted to HTML colors.

```
21 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
22 \begin{group}%
23 \LWR@FBcancel%
```

Use the xcolor package to convert to an HTML color space:

```
24 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Print the converted color:

```
25 \#\LWR@tempcolor%
26 \endgroup%
27 }
```

`\LWR@borderpadding` $\{\langle colorstyle \rangle\} \{\langle color \rangle\}$ 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 \LaTeX 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` $[\langle model \rangle] \{\langle color \rangle\} \{\langle text \rangle\}$

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` $[\langle model \rangle] \{\langle color \rangle\}$

Ignored. Use `\CSSFilename` instead.

```
43 \renewcommand*\LWR@pagecolor}[2][named]{}
```

`\nopagecolor` Ignored.

```
44 \let\nopagecolor\relax
```

`\colorbox` $[\langle model \rangle] \{\langle color \rangle\} \{\langle text \rangle\}$

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` [*⟨framemodel⟩*] {*⟨framecolor⟩*} [*⟨boxmodel⟩*] {*⟨boxcolor⟩*} {*⟨text⟩*}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

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 {% no background color
75   \InlineClass[%
76     \LWR@borderpadding{#1}{#2}%
77   ]{fcolorbox}{#5}%
78 }%
79 {% yes background color
80   \InlineClass[%

```

```

81 \LWR@borderpadding{#1}{#2} ; %
82 background:\LWR@colorstyle{#3}{#4}%
83 ]{fcolorbox}{#5}%
84 }%
85 \endgroup%
86 }

```

`\fcolorboxBlock` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

87 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{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 {% no background color
95   \begin{BlockClass}[%
96     \LWR@borderpadding{#1}{#2}%
97   ]{fcolorboxBlock}
98   #5
99   \end{BlockClass}%
100 }%
101 {% yes background color
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 }

```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the lwarp core: section [68.1](#)

`\LWR@subfcolorminipage` {*framemodel*} {*framecolor*} {*background tag*} {*height*}

```

113 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
114 \begin{BlockClass}[%

```

```

115 #3%
116 \LWR@borderpadding{#1}{#2} ; %
117 \IfValueT{#4}{height:\rndprintlength{\LWR@tempheight} ; }%
118 width:\rndprintlength{\LWR@tempwidth}%
119 ]{fcolorminipage}%
120 }

```

Env fcolorminipage [*1:frame model*] [*2:frame color*] [*3:box model*] [*4:box color*] [*5:align*] [*6:height*]
 [*7:inner-align*] [*8:width*]

```

121 \NewDocumentEnvironment{fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
122 {%
123 \LWR@FBcancel%
124 \setlength{\LWR@tempwidth}{#8}%
125 \IfValueT{#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 decended 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 164 **lwarp-xfrac.sty**

§ 240 Package **xfrac**

Pkg xfrac Supported by adding xfrac instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

 font size

In the user's document preamble, lwarp should be loaded after font-related setup. During HTML conversion, this font is used by lwarp to generate its initial PDF output containing HTML tags, later to be converted by `pdftotext` to a plain text file. While the text may be in any font which `pdftotext` can read, the math is directly converted into SVG images using this same user-selected font. `xfrac` below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine `\xfracHTMLfontsize` with a different em size.

`\sfrac` [*instance*] {*num*} [*sep*] {*denom*}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. `\scalebox` is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a `lateximage`, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

`\xfracHTMLfontsize` User-redefinable macro which controls the font size of the fraction.

```
3 \newcommand*\xfracHTMLfontsize{.6em}
```

```
4 \end{warpall}
```

for HTML output: 5 \begin{warpHTML}

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*\LWR@htmlsmallfontstart{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"}{}}%
8 \LWR@nestspan%
9 %
10 }
11
```

```

12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/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
27 \DeclareInstance{xfrac}{lmr}{text}{
28 numerator-format = {%
29 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
30 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
31 denominator-format = {%
32 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
33 \LWR@htmlsmallfontstart{ }\,#1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

34 scaling = false
35 }
36
37 \DeclareInstance{xfrac}{lmss}{text}{
38 numerator-format = {%
39 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
40 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
41 denominator-format = {%
42 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
43 \LWR@htmlsmallfontstart{ }\,#1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```

44 scaling = false
45 }
46
47 \DeclareInstance{xfrac}{lmtt}{text}{
48 numerator-format = {%
49 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
50 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
51 denominator-format = {%
52 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
53 \LWR@htmlsmallfontstart{\,\,#1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```

54 scaling = false
55 }

56 \end{warpHTML}

```

File 165 **lwarp-xltxtra.sty**

§ 241 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 \DeclareDocumentCommand{\showhyphens}{m}{-}

```

File 166 **lwarp-xmpincl.sty**

§ 242 Package **xmpincl**

Pkg xmpincl Emulated.

for HTML output: Discard all options for lwarp-xmpincl:

```

1 \LWR@ProvidesPackageDrop{xmpincl}

2 \newcommand*{\includexp}[1]{}


```

File 167 **lwarp-xtab.sty**

§ 243 Package **xtab**

Pkg xtab xtab is emulated during HTML output.

for HTML output: 1 \LWR@ProvidesPackageDrop{xtab}

 **misplaced alignment** For \tablefirsthead, etc., enclose them as follows:
alignment tab character &

```

\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 168 **lwarp-zwpage layout.sty**

§ 244 Package **zwpage layout**

Pkg zwpage layout zwpage layout is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{zwpage layout}

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{rgbbblue}{rgb}{0,0,1}
17%   \ifzwpl@redefinetcmyk
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#{\gdef\ZW@oddwarning}
40 \def\SetEvenPageMessage#{\gdef\ZW@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

§ 244 Change History

v0.10		Docs: Table: Float data structures. 320
General: 2016/03/08 Initial version . . .	1	Docs: Trademarks section. 394
v0.11		Docs: Troubleshooting
General: 2016/03/11	1	cross-references. 106
Added section: Operating-System		Test Suite: Assigned cleveref name
portability.	114	for Test Float. 1
Added section: Selecting the		Test Suite: Floatrow 1
operating system.	71	v0.15
Test Suite: limages and index in		General: 2016/04/06 1
README.txt	1	Added 436
Test Suite: MS-Windows in		Ampersand (&): Fixed handling
README.txt	1	when passed as an argument. . . 271
v0.12		Docs: Added warning icons for
\LWR@newhtmlfile: Bugfix: TOC		items needing special attention. 111
with numbered files.	226	Docs: Clarify print/HTML output. 72
General: 2016/03/14	1	Docs: Moved the supported
Global: Uses \p@(type) in float		functions table to the
captions.	1	introduction. 37
Test Suite: Sub-figures	1	Files: lwarp_formal.css added. . . . 1
v0.13		Fix: steps counter 436
\CaptionSeparator: Fix for newer		Fixed & handling. 434
babel package.	322	Test Suite: test_suite_formal.css
\LWR@LwarpStart: \up and \fup . .	242	file added. 1
General: 2016/03/24	1	v0.16
Removed package: subfig	1	General: 2016/04/11 1
Test Suite: Ordinals, Subcaption . .	1	\titlingpage: Improved
tikzpicture: Fix dollar-redefined		print-output spacing. 250
bug for newer package.	540	Added XeLaTeX, LuaLaTeX
v0.14		support. 124
\LWR@htmlsectionfilename: Fix:		Docs: Font and UTF-8 support. . . 69
Links to home page.	196	Docs: Moved location of
General: 2016/03/31	1	\usepackage{lwarp}. 70
floatrow: Added.	431	Docs: Text not converting. 106
Docs: Commands for a successful		Fix: amsmath options clash 128
HTML conversion.	76	Fix: newtxmath compatibility. . . 128
Docs: Commands into a warpprint		Lwarp no longer selects fonts. 69, 123
environment.	73	Removed package: suffix 1
Docs: Newclude limitations. . . .	97	Test Suite: Improved titlingpage. 250
Docs: Table: Cross-referencing		Test Suite: Lwarp no longer selects
data structures.	310	fonts. 1

Test Suite: Supports XeLaTeX, LuaLaTeX.	1	verse: Supports verse, memoir packages.	563
xfrac: Adjusted for the use of any font:	575	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage.	371
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page.	196	\HTMLFilename: Docs: Escape filename underscores.	195
General: 2016/04/14	1	\HomeHTMLFilename: Docs: Escape filename underscores.	196
mdframed: Added.	474	\LWR@LwarpStart: Enabled \ equal to \newline.	241
Test Suite: Fix: Print-version front-matter page numbers.	1	\LWR@doubledollar: MathJax support.	338
Test Suite: Mdfamed	1	\LWR@filestart: lwarp_mathjax.txt loaded.	239
v0.18		\LWR@hspace: Fix: \hspace length computations.	386
\LWR@hspace: \hspace supported.	386	\LWR@minipagestartpars: Surpresses paragraph tags between minipages.	384
\LWR@includegraphicsb: Add: svgz file extension.	451	\LWR@singledollar: MathJax support.	338
em, ex, %, px dimensions preserved.	451	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size.	354
Fix: \linewidth, \textwidth, \textheight inside a minipage.	451	\minipagefullwidth: Added: No width tag for the next minipage in HTML.	371
Improved HTML output linebreaks.	451	\rule: Added	390
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic.	326	\warpHTMLonly: Added.	118
\LWR@newhtmlfile: sideroc after title, improving responsive design.	225	\warpprintonly: Replaces \rowprintedonly.	118
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic.	244	\xfracHTMLfontsize: Added.	574
\LWR@subhyperref: Improved HTML output linebreaks.	318	General: 2016/06/08	1
\LWR@subhyperrefclass: Improved HTML output linebreaks.	318	Added optional args.	488
\LWR@subinlineimage: Surpress extra space.	319	Avoids MathJax.	337
General: 2016/05/19	1	cleveref: Loaded \AtEndPreamble.	367
File: lwarp.css: Improved TOC outline display.	1	CSS for table note item.	539
Files: lwarp.css and lwarp_formal.css: Improved responsive design.	1	Docs: Math options.	70
Microtype disabled during HTML generation	124	Docs: Table: Cross-referencing data structures, updated.	310
PDF Unicode input characters.	112	File: lwarp_mathjax.txt added.	1
Test Suite: Verse package	1	File: lwarp.css: tnoteitemheader added	1
lateximage: pdfcrop: --hires added.	356	Introduction: MathJax support mentioned.	36
Reorganize \HomeHTMLFilename logic.	356	MathJax support added.	339, 345, 346
Surpress extra space.	356	Options: mathsvg and mathjax	116

Supports colored <code>\rule</code>	569	<code>\LWR@subhtmlclass</code> :	
<code>titles: null \pagestyle</code> and		Factored code.	206
<code>\thispagestyle</code> for HTML. . .	540	<code>\SetHTMLFileNumber</code> : Add: Control	
<code>tikzpicture</code> : Adapts to tikz version. .	540	file numbers.	196
<code>equation</code> : MathJax support.	344	<code>\cpagerefFor</code> : User-redefinable	
v0.20		word for page references.	367
<code>\BlockClassSingle</code> : Renamed from		<code>\dotfill</code> : Inserts an ellipsis.	383
" <code>LWR@htmldivclassline</code> ".	208	<code>\hfill</code> : Inserts a <code>\quad</code>	383
<code>\HTMLDescription</code> : Added		<code>\hrulefill</code> : Inserts a short rule. . .	383
<code>\NewHTMLdescription</code> .		<code>\hyperindexref</code> : Print mode	
(Renamed in v0.30.)	218	provided in case <code>hyperref</code> not	
<code>\HTMLFilename</code> : No longer escape		used.	334
<code>underscores</code>	195	<code>\pageref</code> : Added.	317
<code>\HomeHTMLFilename</code> : No longer		<code>\tracinglwrap</code> : Added.	133
escape <code>underscores</code>	196	General: 2017/02/09	1
<code>\InlineClass</code> : Renamed from		<code>afterpage</code> : Added.	398
" <code>inlineclass</code> ".	209	<code>alltt</code> : Added.	399
<code>\LWR@LwrapStart</code> : Fix: math cross		<code>bookmark</code> : Added.	407
<code>references</code>	242	caption and subcaption supported. .	1
<code>\LWR@closeparagraph</code> : <code>\unskip</code>		<code>cleveref</code> and referencing patches:	
extra spaces.	212	Applied <code>\AfterEndPreamble</code> . . .	367
No break tags in the start/end of a		<code>draftwatermark</code> : Added.	412
<code>tabular</code>	212	<code>eso-pic</code> : Added.	417
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	384	<code>everypage</code> : Added.	417
<code>\LWR@filestart</code> : Adds meta		<code>extramarks</code> : Added.	418
<code>description</code>	239	<code>fancyhdr</code> : Added.	422
<code>\LWR@hspace</code> : Add: Supports HTML		<code>hyperref</code> : Additional user macros. .	457
thin breakable space.	386	<code>keyfloat</code> : Added.	463
<code>\LWR@htmldivclass</code> : Added		<code>letterspace</code> : User-interface	
optional style.	207	emulated.	465
<code>\LWR@htmlclass</code> : Added		<code>listings</code> : Added.	467
optional style.	206	<code>lrcaption</code> : Added.	472
<code>\LWR@htmlsectionfilename</code> :		<code>lwrap-newproject</code> : Added.	138
<code>HTMLFilename</code> : removed		<code>microtype</code> : User-interface	
additional trailing <code>'</code> , and may be		emulated.	483
empty.	196	<code>needspace</code> : Added.	491
Sections called "Index" or "index"		<code>nowidow</code> : Added.	493
have an underscore prepended to		<code>placeins</code> : Added.	506
their filenames if no prefix. . .	196	<code>ragged2e</code> : Added.	508
<code>\LWR@includegraphicsb</code> : Fix:		<code>setspace</code> : Improved support. . .	514
<code>\linewidth</code> in a floatrow.	451	<code>textpos</code> : Added.	534
Fix: Expands filename.	451	<code>titles</code> : Added.	540
<code>\LWR@longtablecaptiontag</code> :		<code>titlesec</code> : Added.	543
Fix: Pairs in captions.	296	<code>titletoc</code> : Added.	545
<code>\LWR@section</code> : Combined		<code>titling</code> : Improved compatibility. .	546
higher-level sections together		<code>tocloft</code> : Added.	553
into files.	231	<code>wallpaper</code> : Added.	564
<code>\LWR@setOSWindows</code> : Auto-detects		<code>wrapfig</code> : Added.	565
operating system.	114	Added <code>@</code> , <code><</code> , <code>></code> columns.	267

Added single-expansion data arrays.	194	lwarpmk: Fix: lwarpmk limages for Windows.	179
Code factored into independent lwrap_html files.	394	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script.	179
Docs: Examples for generating HTML file names.	62	Add: Errors for misplaced packages.	119
Docs: Improved index.	1	Docs: Added internet class. . . .	41
Enhanced titling support.	248	Docs: Added TeX2page, GladTeX. .	41
File: lwrap.css: Minor fixes for validation.	1	Docs: Installing on Windows. . . .	47
File: lwarpmk used to compile print, HTML, indexes, and lateximages.	1	File lwrap_tutorial.txt added. . .	51
Fix: \linewidth in a floatrow. . .	434	v0.22	
Improved float caption type handling.	429	\LWR@parseDcolumn: Added tabular D column.	279
Moved sidebar and example code to test suite.	1	\LWR@parsebangcolumn: Added tabular ! column.	276
Page geometry set to 6in wide with large margins.	125	\LWR@parsetablecols: Unknown table column types become 1. Added tabular D, !, X columns. . . .	280
Parallel versions of aux files for print/HTML.	1	\LWR@printmccoldata: Added tabular D, !, and X columns. . .	292
Removed reliance on make, grep, gawk.	1	General: 2017/03/02	1
Tabular: \unskip extra spaces. . .	267	abstract: Added.	396
Test Suite: HTML meta descriptions.	1	change page: Added.	409
Verbatim: Added.	424	dcolumn: Added.	412
verbatim: Added.	259	fnright: Added.	444
BlockClass: Added optional style. .	208	geometry: Nullified commands. .	445
Renamed from "blockclass". . . .	208	indentfirst: Added.	463
LWR@nestspan: Fix: Minipages inside a span.	203	layout: Added.	464
v0.21		lscape: Added.	472
\LWR@LwrapStart: Changed lateximages to a .txt file. . . .	241	mcaption: Added.	474
\LWR@filestart: Skip title if not given.	239	nameref: Added.	491
\LWR@newhtmlfile: Skip title if not given.	225	nextpage: Added.	492
\marginpar: Fixed source listing. . .	222	parskip: Added.	505
General: 2017/02/23	1	showkeys: Added.	516
fontenc: Added.	438	sidecap: Added.	516
fontspec: Added.	439	tabularx: Added.	529
inputenc: Added.	463	varioref: Supported.	82
newclude: Added.	491	verse: Added.	562
newunicodechar: Added.	492	v0.23	
lwarpmk: Fix: lwarpmk again for Windows.	179	\LWR@parsetablecols: Fix for vert bar column type.	280
		\LWR@printmccoldata: Fix for vert bar column type.	292
		General: 2017/03/02	1
		v0.24	
		\LWR@hspace: Add: \hspace \fill converts to 2em	386
		\LWR@htmlfileref: Fix: Index links while \tracinglwrap.	313

\hypertocfloat: List of floats responds to lofdepth, lotdepth.	332	sidenotes: Added.	517
General: 2017/03/15	1	lwarp.css: Improved responsive marginpar and marginblock. . .	140
floatrow: Support for subfig. . . .	431	Docs: Improved MiKTeX install instructions.	44, 47
subfig: Added.	521	Dollar span avoided in a lateximage.	337
tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary.	539	Footnotes now are \LaTeX boxes instead of pagenotes.	218
Docs: Filename underscore. . .	60, 77	lateximage: Labels track page numbers of lateximages.	356
No longer preloads subcaption; conflicted with subfig.	127	Print mode now uses a minipage of \linewidth.	356
picture: Fix for inline images. . . .	368	picture: Fix for \makebox in picture.	368
tikzpicture: Fix for inline images. .	540		
v0.25		v0.27	
\LWR@loadnever: Added the ability to prevent conflicting packages. .	120	\LWR@footnotetext: Fix for table footnote par tags.	220
\addcontentsline: Handles theorems.	325	General: 2017/04/04	1
General: 2016/03/22	1	lettrine: Added.	465
amsthm: Added.	400	microtype: Fix with Xe \LaTeX , Lua \LaTeX	483
ccaption: Prevented.	409	soul: Added.	519
ellipsis: Added.	413	ulem: Added.	561
emptypage: Added.	414	Docs: Installing utilities for MacOS.	49
framed: Added.	441	Docs: Limitations of saveboxes. .	78
lips: Added.	466	Page geometry modified to reduce line overflow.	125
mdframed: Help avoid hyphenation.	475	v0.28	
ntheorem: Added.	493	\@wrindex: Improved indexing. . . .	334
showidx: Added.	516	\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.)	218
theorem: Added.	534	\LWR@LwarpEnd: If FormatEpub or FormatWordProcessor, no bottom nav.	244
Basic \LaTeX theorems: improved css.	259	\LWR@LwarpStart: FormatWordProcessor forces single-file output.	241
Docs: Adds credits for patched code.	1	\LWR@filestart: Adds HTML meta author.	239
Docs: Testing lwarp.	104	\LWR@forcenewpage: Forces new PDF page before major environments.	200
Fix: Allows Xe \LaTeX and Lua \LaTeX to preload graphics and graphicx. .	120	\LWR@htmlcomment: Breaks ligatures in HTML comments.	205
v0.26		\LWR@includegraphicsb: Adapts to graphics syntax.	451
General: 2017/03/31	1		
cutwin: Added.	411		
endnotes: Added.	414		
floatflt: Added.	430		
footmisc: Added.	439		
footnotehyper: Added.	441		
footnote: Added.	440		
marginfix: Added.	473		
marginnote: Added.	473		
mparhack: Added.	486		
pagenote: Supported as-is.	504		

\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav.	225	\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	216
\LWR@parsetablecols: Added L, C, R, J column types.	280	\HTMLLanguage: Renamed from \MetaLanguage.	238
\LWR@startref: Removed space. . .	315	\HTMLPageBottom: Renamed from \SetPageBottom.	217
\chapter: If EPUB, prints footnotes before each section.	236	\HTMLPageTop: Renamed from \SetPageTop.	216
\hyperindexref: Improved indexing.	334	General: 2017/04/29	1
\textup: Fixed span class.	379	lwarmp-newproject removed, and combined with lwarmp.	138
General: 2017/04/14	1	lwarmpmk: Add: xdyfile configuration option.	179
glossaries: Added.	446	lwarmpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex.	179
graphics: Added.	447	lwarmpmk: Fix: xindy now used for print index generation with latexmk.	179
tabularx: Fix for optional pos. . .	529	lwarmpmk: language now used for both index and glossary generation.	179
tabulary: Added.	530	File: lwarmp_html.xdy renamed to lwarmp.xdy.	176
lwarmpmk: Add: printglossary and htmlglossary commands. . .	179	Fix: *.css files only written in print mode.	140
Added boolean FormatEPUB. . . .	134	Fix: lwarmp.xdy only written in print mode.	176
Added boolean FormatWordProcessor.	134	Fix: lwarmp_mathjax.txt only written in print mode.	177
Added boolean HTMLDebugComments.	133	Option OSWindows replaces macro \warpOSWindows.	117
Added boolean HTMLMarkFloats	134	Option latexmk replaces macro \UseLatexmk.	117
Docs: Modfying lwarmpmk and index processing.	104	Option lwarmpmklang changed to IndexLanguage.	116
File lwarmp_mathjax.txt: Updated CDN repository.	177	Option xdyFilename added. . . .	116
Forced oneside to maintain large right margin.	125	Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename.	117
v0.29		v0.31	
\LWR@includegraphicsb: Fix: Error when no optional arguments. . .	451	General: 2017/05/15	1
General: 2017/04/15	1	keyfloat: Improved compatibility. .	463
lwarmpmk: Add: language option for config files.	179	v0.32	
Add: lwarmpmklang option for lwarmp.	116	\RequirePackage: Fix: Ignores blanks in package list.	129
Docs: Using a glossary	67	General: 2016/06/09	1
File *.lwarmpmkconf: Add: language option for config files. .	139	glossaries: Prevent error with \glo@name not defined.	335
File lwarmpmk.conf: Add: language option for config files. .	139		
v0.30			
\CSSFilename: Renamed from \NewCSS.	217		
\HTMLAuthor: Renamed from \HTMLauthor.	218		
\HTMLDescription: Renamed from \NewHTMLdescription.	218		

lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes.	179	\LWR@nullfonts: Improved font control.	380
v0.33		\LWR@restoreorigformatting: booktabs: Works inside lateximage.	340
\HTMLAuthor: Fix: Provides empty default author if none given. . .	218	Improved font control.	340
\LWR@loadbefore: Fix: No \PackageError if already loaded.	119	\LWR@subhtmlclass: Moved optional argument in front of mandatory.	206
\LWR@parseatcolumn: Fix: Column alignment with leftmost @.	276	\LWR@tabledatacolumn: booktabs: Works inside lateximage.	303
\LWR@tabledatasinglecolumn: Fix: Macros in tabular could cause extra data cell.	286	\fboxBlock: Added.	375
\LWR@vspace: Add: \vspace nullified.	387	\makebox: Fix: Handles paren arg. .	373
\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &.	271	General: 2017/08/08	1
General: 2017/07/10	1	babel-french: Adds fixed-width HTML spaces to punctuation. .	201
amsmath: Removed fleqn option.	128	balance: Added.	405
fancyhdr: Fix: Optional args for \lhead, etc.	422	booktabs: Works inside lateximage.	305, 407
Add: Tabular at and bang columns now have their own HTML columns.	267	boxedminipage2e: Added.	408
cleveref: Fix: Loaded \AtEndPreamble.	367	boxedminipage: Prevented.	408
Fix: Incorrectly-inline math environments.	346	crop: Added.	410
New handling of & to localize catcode changes.	267	enumerate: Added.	415
v0.34		enumitem: Added, no longer required.	415
\@fnsymbol: Text symbols instead of math.	251	everyshi: Added.	418
\InlineClass: Moved optional argument in front of mandatory.	209	fancybox: Added.	420
\LWR@htmldivclass: Moved optional argument in front of mandatory.	207	fancyvrb: Added, no longer required.	423
\LWR@htmlclass: Moved optional argument in front of mandatory.	206	figcaps: Added.	428
\LWR@htmlclassline: Moved optional argument in front of mandatory.	207	filecontents: Required. Patched for morewrites.	126
\LWR@htmlspanclass: Moved optional argument in front of mandatory.	204	floatpag: Added.	431
		flushend: Added.	437
		fullpage: Added.	444
		hyperxmp: Added.	462
		idxlayout: Added.	462
		marginfit: Added.	473
		mdframed: Improved mdtheorem patch.	480
		moreverb: Added.	484
		paralist: Added.	504
		pdfscape: Added.	505
		pdfsync: Added.	505
		prelim2e: Added.	506
		rotfloat: Added.	512
		savetrees: Added.	513
		shadow: Added.	515
		syntonly: Added.	528

titles: No longer required.	540	super/subscripts in a <code>lateximage</code>	340
titleref: Prevented.	543	<code>\LWR@section</code> : Improved spacing.	231
xmpinl: Added.	577	<code>\LWR@stoppars</code> : Extra HTML source space after paragraphs.	214
Added.	572	<code>\fbox</code> : Fix: Uses <code>\fboxrule</code> and <code>\fboxsep</code>	375
Docs: Horizontal space limitations.	1	<code>\framebox</code> : Fix: Handles width and horiz position.	374
Docs: Misplaced alignment character.	106	<code>\makebox</code> : Fix: Handles width and horiz position.	373
File: README.txt: updated.	1	General: 2017/08/17	1
File: <code>lwarp_mathjax.txt</code> version change.	177	<code>babel-french</code> : Adjustements for French variants, load order, footnotes, ellipses.	201
Fix: Added the <code>eqnarray</code> environments.	346	footnote: Extra HTML source space after paragraphs.	440
Improved font control.	378	<code>siunitx</code> : Fix for <code>babel-french</code>	361
Lists refactored to remove <code>enumitem</code> requirement.	261	<code>siunitx</code> : Improved symbol support.	519
Verbatim refactored to remove <code>fancyvrb</code> requirement.	257	transparent: Added.	558
<code>lateximage</code> : Fix: <code>lateximage</code> with <code>minipage</code> , <code>\parbox</code> , <code>\makebox</code> , <code>\fbox</code> , <code>\framebox</code> , <code>\raisebox</code> , <code>\scalebox</code> , <code>\reflectbox</code>	356	<code>upref</code> : Added.	562
BlockClass: Moved optional argument in front of mandatory.	208	<code>xcolor</code> : Added <code>\fcolorboxBlock</code> , <code>\colorboxBlock</code>	567
<code>fminipage</code> : Added.	376	<code>xcolor</code> : Fix: Background none in print mode.	567
<code>LWR@nestspan</code> : Fix: Minipages, BlocksClass, and lists inside a span.	203	<code>xcolor</code> : Refactored <code>\LWR@colorstyle</code>	569
<code>LWR@tabular</code> : <code>booktabs</code> : Works inside <code>lateximage</code>	306	<code>xcolor</code> : Uses <code>\fboxrule</code> and <code>\fboxsep</code>	567
v0.35		<code>xcolor</code> : <code>\fcolorbox</code> etc. now work inside <code>lateximage</code>	567
General: 2017/08/08	1	Docs: Reorganized: Special cases and limitations.	76
Fix: <code>\textbf</code> and related.	378	Source: Improved formatting.	1
v0.36		<code>lateximage</code> : Footnotes appear in regular text instead of the <code>lateximage minipage</code>	356
<code>\LWR@HTMLsanitize</code> : Fix for <code>babel-french</code>	355	<code>LWR@tabular</code> : Fix for <code>babel-french</code>	306
<code>\LWR@HTMLsanitizeexpand</code> : Fix for <code>babel-french</code>	355	v0.37	
<code>\LWR@closeparagraph</code> : Extra HTML source space after paragraphs.	212	<code>\include</code> : Maintains independent .aux files for HTML.	131
<code>\LWR@currenttextcolor</code> : Fix for <code>\rule</code> when <code>xcolor</code> not loaded.	388	General: 2017/08/19	1
<code>\LWR@footnotetext</code> : Extra HTML source space after paragraphs.	220	\TeX accents: Added.	136
Force HTML superscripts.	220	<code>babel-french</code> : Adjustment for load order.	201
<code>\LWR@nullfonts</code> : Fix: Filenames while using MathJax.	380	color: Prevented.	410
<code>\LWR@restoreorigformatting</code> : <code>siunitx</code> : Improved		comment: Maintains independent cutfiles for print, HTML.	118
		<code>siunitx</code> : Improved symbol support.	519
		textcomp: Improved support.	531

lwarpmk: Removes additional HTML aux files.	179	Supports authblk with <div>s of class oneauthor instead of tabular.	252, 549
File handles reorganized.	131	\AddSubtitlePublished: Added. . .	254
v0.38		\LWR@domulticolumn: Add: Optional vpos and # rows.	294
\@seccntformat: Added for appendix.	231	\LWR@restoreorigformatting: Appended with \appto instead of calling various macros.	340
\ForceHTMLPage: Added.	228	\LWR@tabledatacolumnntag: Don't start a data cell if see \TabularMacro.	303
\ForceHTMLTOC: Added.	229	\ResumeTabular: Added.	302
\LWR@section: \part* starts a new HTML page, for appendix.	231	\TabularMacro: Added.	302
Modified spacing, uses \numberline.	231	\multicolumnrow: Added.	300, 490
\numberline: Added trailing \quad.	330	\printauthor: Removed minipages.	249
\part: Fix with article class.	236	Supports authblk with <div>s of class oneauthor instead of tabular.	249
General: 2017/08/27	1	\thanksmarkseries: Removed minipage footnotes.	550
appendix: Added.	403	General: 2017/09/05	1
arabicfront: Added.	404	a4wide: Added.	395
caption2: Prevented.	409	a4: Added.	395
chappg: Added.	410	a5comb: Added.	395
color: Forces xcolor as well.	410	addlines: Added.	398
fix2col: Added.	429	anysize: Added.	403
fncychap: Added.	437	authblk: Added.	404
grffile: Added.	457	bigdelim: Added.	406
metalogo: Added.	483	bigstrut: Added.	407
nonumonpart: Added.	492	chngepage: Prevented.	409
nopageno: Added.	493	ebook: Added.	413
pagenote: Option page disabled.	504	fullwidth: Added.	445
realscripts: Added.	509	supertabular: Added.	527
relsize: Added.	510	textarea: Added.	530
romanbarpagenumber: Added. . .	511	titling: Improved compatibility. .	546
romanbar: Added.	511	titling: Removed extraneous center environments.	547
scalefnt: Added.	513	typearea: Added.	560
siunitx: Removed from lwarp core.	519	xtabular: Added.	577
textcomp: Removed from lwarp core.	531	zwpagelayout: Added.	579
tocbibind: Added.	552	Add: New optional vpos argument.	488
xltxtra: Added.	576	Add: Supports left/right border for bigdelim.	488
lwarpmk: Added print1 and html1 actions.	179	Docs: Reorganized tabular discussion.	92
Added \markboth, \sloppy, etc.	199	Fix: Long text argument.	488
Docs: Enhanced <i>Supported</i> <i>Functions</i> table.	37	Titlepage \published and \subtitle removed.	
Docs: Index, tocbibind.	86		
Docs: Starred sections.	83		
v0.39			
\@maketitle: titling version.	549		
Native L ^A T _E X version.	252		
Removed minipages.	252, 549		

\AddSubtitlePublished		quotchap: Added.	507
restores.	253	sectsty: Added.	513
titlepage: Clear pending footnotes.	248	stabular: Added.	521
Removed minipages.	248	tabs: Added.	529
titlingpage: Clear pending		textcomp: Additional symbols,	
footnotes.	547	improved XeLaTeX and LuaLaTeX	
v0.40		support.	531
\@chapcntformat: Added for		tocbibind: Improved for	
tocbibind, anonchap.	231	\simplechapter.	552
\LWR@hline: Added.	306	xltxtra: Fix for \showhyphens with	
\LWR@includegraphicsb: Add: Full		XeLaTeX.	576
\graphicspath support.	451	Improved bigdelim borders.	488
\LWR@nullfonts: Fix: Long		No longer preloads xfrac.	128
arguments for expandable			
command.	380	v0.41	
\LWR@restoreorigformatting:		\LWR@addcmidruletrim: Add:	
Improved \TeX logos inside a		\cmidrule trims.	290
lateximage.	340	\LWR@clearmidrules: Add:	
Improved symbols inside a		\cmidrule trims.	289
lateximage.	340	\LWR@closetabledatacell: Add:	
Nullified \InlineClass, etc.		Mute > for \bottomrule.	269
inside a lateximage.	340	Fix: At/bang column with	
\LWR@tabledatacolumntag: Fix for		\multirow.	269
bigdelim: \ldelim, \rdelim.	303	Fix: Cancel < for \multicolumn.	270
\chapter: Added support for		\LWR@domulticolumn: Add:	
quotchap.	236	\cmidrule trims.	294
\multicolumnrow: Fix: Adapts to		Added vertical rules.	295
older multirow and xparse.	300	\LWR@nullifyNoAutoSpacing: Fix:	
\simplechapterdelim: Added for		\NoAutoSpacing in a tabular	
tocbibind, anonchap.	231	with babel-french.	306
\underline: Added.	382	\LWR@parsebarcolumn: Added	
General: 2017/09/25	1	vertical rules.	278
adjmulticol: Added.	397	\LWR@printatbang: Add: \cmidrule	
anonchap: Added.	403	trims.	285
bigdelim: Improved		Add: Mute at and bang columns	
documentation.	406	for \bottomrule.	285
cuted: Added.	411	\LWR@printbartag: Added vertical	
dblfnote: Added.	412	rules.	285
fnpos: Added.	438	\LWR@subaddcmidruletrim: Added.	290
graphics: Moved out of the lwarp		\LWR@subcmidrule: Add: \cmidrule	
core.	447	trims.	289
graphics: Restores		\LWR@tabledatasinglecolumntag:	
\includegraphics and		Add: \cmidrule trims.	286
\DeclareGraphicsExtensions		Add: Mute < for \bottomrule.	286
in a lateximage.	447	\LWR@tabularfinishrow:	
graphicx: Moved out of the lwarp		Unfinished tabular rows	
core.	457	automatically filled.	272
grffile: Directly supported.	457	\mcolrowcell: Added for	
midfloat: Added.	484	\multicolumnrow cells.	305
pfnote: Added.	506	General: 2017/10/07	1
		Add: \cmidrule trims.	488

Added vertical rules.	489	LWR@tabular: Fix: \NoAutoSpacing in a tabular with babel-french. . .	307
Fix: < spec.	489	Improved rules.	308

§ 244 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	A
\\$	337
\&	195, 4859
\(.	6230
\)	6230
\,	76
\@setcpageref	6954
\@setcref	6950
\@setcrefrange	6951
\@author	246
\@begintheorem	4689
\@capttype	5919
\@chapcntformat	4066
\@currentlabelname	5699
\@date	246
\@dlbfloat	5897
\@endtheorem	4699
\@ensuredmath	6237
\@float	5897
\@fnsymbol	4511
\@footnotetext	3835
\@include	450
\@makecaption	5921
\@makefnmark	3807
\@makefntext	3806
\@maketitle	49, 4534
\@mpfootnotetext	3836
\@opargbegintheorem	4694
\@seccntformat	4064
\@title	246
\@wrglossary	6171
\@wrindex	6164
\\	384
\$	338
\$\$	338
~	76
a4 (package)	395
a4wide (package)	395
a5comb (package)	395
abstract (environment)	4597
abstract (package)	395
\abstractname	62, 4596
accents	228
\addcontentsline	5973
addlines (package)	398
\AddSubtitlePublished	4559
adjmulticol (package)	397
Adobe (program)	43
\affiliation	4421
afterpackage (package)	125
afterpage (package)	398
algorithmic	
with newfloat, trivfloat	560
algorithmicx (package)	399
align (environment)	6564
align* (environment)	6587
alignment tab character &, misplaced	92, 264, 271
alltt (package)	399
\AmS	7473
AMSMath	
split miss-numbered	87, 89, 335, 494
amsmath (package)	128
amsthm (package)	400
\and	246
anonchap (package)	402
anysize (package)	403
appendix (package)	403
arabicfront (package)	404
array (package)	310
AsciiDoc (program)	42
AsciiDoctor (program)	42

Asciidoctor-LaTeX (program)	42	LWR@tableparcell	268
\attrib	97, 256, 563	LWR@tabularmutemods	303
\attribution	4607	LWR@tracinglwarp	133
authblk (package)	404	LWR@validtablecol	280
author		LWR@verbtags	258
HTML meta tag	74, 218	mathjax	115
\author	74, 246	usingOSWindows	114
autosec	230	warpingHTML	115
		warpingprint	115
B		boxedminipage (package)	408
babel (package)	98	boxedminipage2e (package)	408
\backmatter	4048	\boxframe	137
balance (package)	405	bugs	106
\BaseJobname	3252	BVerbatim (environment)	183
BaseJobname (option)	116		
baseline		C	
tabular	280	calc (package)	127
\bfseries	7269	Calibre	99
\BibTeX	7468	caption (package)	127
bigdelim (package)	405	caption2 (package)	409
bigstrut (package)	407	\caption@begin	5952
BlockClass (environment)	3581	\caption@end	5952
\BlockClassSingle	3593	\captionlistentry	5956
bookmark (package)	407	\captionof	5986
booktabs (package)	407	\CaptionSeparator	5920
booleans:		ccaption (package)	409
CombineHigherDepths	61, 223	center (environment)	6799
FileSectionNames	62, 196	\centering	242
FormatEpub	99, 134	changepage (package)	409
FormatWordProcessor	101, 134	chappg (package)	410
HTMLDebugComments	133	\chapter	4213
HTMLMarkFloats	102, 134	chnpage (package)	409
LWR@amsmultiline	345	\citetitle	4614
LWR@doingapar	210	class:	
LWR@doingcmidrule	268	internet	41
LWR@doinghline	268	cleveref (package)	366
LWR@doingstartpars	210	cmap (package)	70
LWR@doingtbrule	268	\color	570
LWR@emptyatbang	269	color (package)	410
LWR@exitingtabular	303	\colorbox	45
LWR@freezethisfloat	322	\colorboxBlock	55, 6846
LWR@infloatrow	451	CombineHigherDepths (boolean)	61, 223
LWR@intabularmetadata	269	comment (package)	118
LWR@minipagefullwidth	371	Computer Modern	69
LWR@minipagethispar	371	\ConTeXt	7465
LWR@skipatbang	268	counters:	
LWR@skippingmcolrowcell	268	FileDepth	61, 223
LWR@skippingmrowcell	268	lofdepth	331
LWR@starredlongtable	296	lotdepth	331
LWR@startedrow	268	LWR@externalfilecnt	337

LWR@htmlfilenumber	196	ellipsis (package)	413
LWR@latestautopage	323	\emph	7192
LWR@lateximagedepth	354	emptypage (package)	414
LWR@lateximagenumber	354	\end@dlbfloat	5911
LWR@LIpage	354	\end@float	5911
LWR@midrulecounter	288	\EndDefiningTabulars	4853
LWR@minipagedepth	369	endnotes (package)	414
LWR@nextautofloat	323	\enlargethispage	7390
LWR@nextautopage	323	\enskip	385, 7351
LWR@nextequation	339	enumerate (environment)	4725
LWR@prevFileDepth	231	enumerate (package)	415
LWR@spandepth	210	enumitem (package)	415
LWR@tablecolspos	275	environ (package)	127
LWR@tablecolswidth	275	environments:	
LWR@tabletotalcols	275	abstract	4597
LWR@tabletotalcolsnext	275	align	6564
LWR@thisfloat	322	align*	6587
SideTOCDepth	61, 329	BlockClass	3581
tocdepth	60	BVerbatim	183
\cpagerefFor	6953	center	6799
crop (package)	410	description	4753
cross-references		displaymath	6248
missing or incorrect	106	enumerate	4725
CSS		eqnarray	6420
file selection	75	equation	6371
lwrap.css	75	fcolorminipage	121, 6888
per HTML page	75	flalign	6610
project-specific changes	75	flalign*	6633
\CSSFilename	59, 62, 75, 3777	flushleft	6811
cuted (package)	411	flushright	6805
cutwin (package)	411	fminipage	79, 7140, 7188
		gather	6518
D		gather*	6541
danger icon	111	itemize	4711
\date	74	lateximage	6709
dblfnote (package)	412	longtable	2
dcolumn (package)	412	LWR@nestspan	3425
debugging	106	LWR@tabular	5603
HTML debug comments	133	LWR@print@fminipage	7164
\DeclareGraphicsExtensions	2	math	6247
Deja Vu	69	minipage	7016
description		multline	6469
HTML meta tag	74, 218	multline*	6493
description (environment)	4753	picture	368, 6971
displaymath (environment)	6248	quote	4622
\dotfill	7286	theindex	6142
draftwatermark (package)	412	tikzpicture	8
		titlepage	73, 4438
E		titlingpage	14, 73
ebook (package)	413	Verbatim	13

verbatim	4679	project.css	75
verse	2	project.lwarpmkconf	139
warpall	72, 100	project_html.tex	138
warpHTML	50, 64, 72, 101	sample_project.css	75, 176
warpprint	63, 72, 101	tutorial.tex	51
epigraph (package)	416	FileSectionNames (boolean)	62, 196
EPUB		fix2col (package)	429
conversion software	99	\flagverse	563
HTML conversion settings	99, 134	flalign (environment)	6610
eqnarray (environment)	6420	flalign* (environment)	6633
equation		Flare (program)	43
miss-numbered	87, 89, 335, 494	float (package)	429
equation (environment)	6371	floatflt (package)	430
error messages	106	\floatname	9
eso-pic (package)	417	floatpag (package)	431
etoolbox (package)	113	\floatplacement	12
everyhook (package)	126	floatrow (package)	431
everypage (package)	417	\floatstyle	15
everyshi (package)	418	\flushbottom	3346
expl3 (package)	126	flushend (package)	437
extramarks (package)	418	flushleft (environment)	6811
		flushright (environment)	6805
F		fminipage (environment)	79, 7140, 7188
fancybox (package)	419	fncychap (package)	437
fancyhdr (package)	422	fnpos (package)	438
fancyvrb (package)	423	font	
FAQ	106	bitmapped	69
fbox		Computer Modern	69
around a minipage	79	Deja Vu	69
\fbox	79, 7123	ligatures	69, 70
\fboxBlock	79, 7132, 7187	packages	69
\fcolorbox	67, 6852	selection	69
\fcolorboxBlock	87, 6887	size — lateximage	87, 335, 354
fcolorminipage (environment)	121, 6888	size — math, SVG	87, 335, 354
figcaps (package)	428	size — xfrac	574
file		fontenc (package)	69, 438
inaccessible	61	fontspec (package)	69, 439
underscore	60, 77	footmisc (package)	439
filecontents (package)	126	footnote (package)	440
FileDepth (counter)	61, 223	footnotehyper (package)	441
files:		footnotes	218
glyphtounicode	70	for	
lwarp.css	75, 140	HTML & PRINT	111
lwarp.xdy	75, 176	HTML output	111
lwarp_formal.css	171	PRINT output	111
lwarp_mathjax.txt	177	\ForceHTMLPage	84, 4023
lwarp_sagebrush.css	166	\ForceHTMLTOC	84, 4029
lwarp_tutorial.txt	51	FormatEPUB (boolean)	99, 134
lwarpmk.conf	138	FormatWordProcessor (boolean)	101, 134
lwarpmk.lua	104	\framebox	7102

framed (package)	441	meta tag	
framed objects	79	author	74, 218
FrameMaker (program)	43	description	74, 218
Frequently Asked Questions	106	generator	239
\frontmatter	4045	viewport	240
ftnright (package)	444	tabular column conversion	281
fullpage (package)	444	\HTMLAuthor	63, 74, 3791
fullwidth (package)	444	HTMLDebugComments (boolean)	133
\fup	7282	\HTMLDescription	63, 74, 3797
\fussy	3348	\HTMLentity	3235
G			
gather (environment)	6518	\HTMLFilename	3253
gather* (environment)	6541	HTMLFilename (option)	60, 117
GELLMU (program)	41	\HTMLFirstPageTop	63, 3754
generator		htmlglossary (option) [lwarpmk]	86, 446
HTML meta tag	239	\HTMLLanguage	63, 4272
geometry (package)	125, 445	HTMLleftmargini (length)	97, 256, 257, 563
getttitlestring (package)	126	HTMLMarkFloats (boolean)	102, 134
GladTeX (program)	42	\HTMLPageBottom	63, 246, 3770
glossaries (package)	85, 445	\HTMLPageTop	63, 3762
glossary		\HTMLUnicode	3243
language	67	HTMLvleftskip (length)	97, 256, 257, 563
processing	67	\hyperindexref	6178
glyphtounicode (file)	70	hyperref (package)	317, 457
graphics (package)	361, 447	\hypertoc	6088
graphicx (package)	361, 457	\hypertocfloat	6101
grffile (package)	457	hyperxmp (package)	462
H			
Hevea (program)	41	I	
\hfill	7284	icon	
\HomeHTMLFilename	3254	warning	111
HomeHTMLFilename (option)	60, 117	idxlayout (package)	462
horizontal space	76	ifplatform (package)	113
between minipages	384	images	
\href	5842	graphicx package	361
\hrulefill	7285	in strange places	107
\hskip	76	showing as HTML	57, 107
\hspace	76, 385, 7381	\include	450
HTML		\includegraphics	181
commands for a successful conver-		indentfirst (package)	463
sion	76	InDesign (program)	43
conversion settings	60	index	
debug comments	133	language	67
EPUB	99, 134	placement and toc options	86
word processor	101, 102, 134	processing	55, 56
filename generation	62	UTF-8	70
headings	109	IndexLanguage (option)	67, 86, 116, 446, 551
in place of SVG images	57, 107	\InlineClass	3601
		inputenc (package)	69, 463
		internet (class)	41

item		LibreOffice (program)	42
empty	77, 261	ligatures	69, 70, 124
itemize (environment)	4711	line numbers	111
\itshape	7274	\linebreak	7384
J			
JavaScript		\LinkHome	3304
MathJax	87, 335	Linux (program)	71, 114
K		lips (package)	466
\kern	76	list item, empty	77, 261
keyfloat (package)	463	listings (package)	467
kvoptions (package)	115	\listof	6054
L		lmodern (package)	69
label		lofdepth (counter)	331
in HTML	242	longtable (environment)	2
math environment	345	longtable (package)	470
language		lotdepth (counter)	331
glossary	67	lscap (package)	472
index	67	ltcaption (package)	472
language HTML metadata	238	LuaLaTeX	
\LaTeX	7447	detection	124
LaTeX2HTML (program)	41	file & section names	228
\LaTeXe	7447	\LuaLaTeX	7459
lateximage (environment)	6709	LuaLaTeX (program) [requirement] . . .	45
\LateximageFontSizeName	6667	\LuaTeX	7459
lateximages		lwarp	
font size	87, 335, 354	loading	70
latexmk (option)	60, 117	options	70
LaTeXML (program)	41	lwarp (package)	70
layout (package)	464	lwarp.css (file)	75, 140
lengths:		lwarp.xdy	
HTMLleftmargini	97, 256, 257, 563	customizing	75
HTMLvleftskip	97, 256, 257, 563	lwarp.xdy (file)	75, 176
LWR@minipageheight	369	lwarp_formal.css (file)	171
LWR@minipagewidth	369	lwarp_mathjax.txt (file)	177
LWR@tempheight	386	lwarp_sagebrush.css (file)	166
LWR@tempraise	386	lwarp_tutorial.txt (file)	51
LWR@tempwidth	386	lwarpmk	
\LWR@cmidrulewidth	288	customizing	104
\LWR@heavyrulewidth	288	[lwarpmk]:	
\LWR@lightrulewidth	288	htmlglossary (option)	86, 446
\LWR@thiscmidrulewidth	288	printglossary (option)	86, 446
\VerbatimHTMLWidth	257	lwarpmk (option)	116
vleftmargini	97, 256, 563	lwarpmk (program)	104, 179
vleftskip	97, 256, 563	lwarpmk.conf (file)	138
letltxmacro (package)	113	lwarpmk.lua (file)	104
letterspace (package)	465	\LWR@addcmidruletrim	5241
lettrine (package)	465	\LWR@addcmidrulewidth	5263
		\LWR@addleftmostbartag	5144
		\LWR@addmathjax	6360
		\LWR@addrulewidth	5246
		\LWR@afterendverbatim	4663

LWR@amsmultline (boolean)	345	LWR@getexparray	3231
\LWR@atbeginverbatim	4649	\LWR@getmynexttoken	4786
\LWR@beginhideamsmath	6405	\LWR@heavyrulewidth (length)	288
\LWR@blackborderpadding	7118	\LWR@hidelatexequation	6346
\LWR@botnavigation	3312	\LWR@hline	5581
\LWR@caption@begin	5933	\LWR@hspace	7359
\LWR@caption@end	5943	\LWR@htmlblockcomment	3521
\LWR@clearmidrules	5204	\LWR@htmlblocktag	3529
\LWR@closeparagraph	3669	\LWR@htmlclosecomment	3493
\LWR@closeprevious	3332	\LWR@htmlcomment	3514
\LWR@closetabledatacell	4814	\LWR@htmldivclass	3558
\LWR@cmidrulewidth (length)	288	\LWR@htmldivclassend	3561
LWR@colafterspec	275	\LWR@html element	3571
LWR@colatspec	275	\LWR@html element class	3545
LWR@colbangspec	275	\LWR@html element class end	3550
LWR@colbarspec	275	\LWR@html element class line	3564
LWR@colbeforespec	275	\LWR@html element end	3574
\LWR@copyfile	485	LWR@htmlfilenumber (counter)	196
\LWR@createautosec	4054	\LWR@htmlfileref	5725
\LWR@currentcss	3776	\LWR@htmlmathlabel	6391
\LWR@currenttextcolor	7391	\LWR@htmlmulticolumn	5365
\LWR@descitem	4739	\LWR@html opencomment	3493
\LWR@docmidrule	5229	\LWR@htmlrefsectionfilename	3297
LWR@doingapar (boolean)	210	\LWR@HTMLsanitize	6670
LWR@doingcmidrule (boolean)	268	\LWR@HTMLsanitizeexpand	6691
LWR@doinghline (boolean)	268	\LWR@htmlsectionfilename	3264
LWR@doingstartpars (boolean)	210	\LWR@htmlspan	3466
LWR@doingtbrule (boolean)	268	\LWR@htmlspanclass	3474
\LWR@domulticolumn	5338	\LWR@htmltag	3484
\LWR@doubledollar	6204	\LWR@htmltagc	3415
LWR@emptyatbang (boolean)	269	\LWR@includegraphicsb	100
\LWR@endhideamsmath	6413	\LWR@indexitem	6152, 6156, 6160
\LWR@endminipage	6980	LWR@infloatrow (boolean)	451
\LWR@endoffline	7334	\LWR@instertatbangcols	4806
\LWR@endsubminipage	7001	LWR@intabularmetadata (boolean) ..	269
\LWR@ensuredoingapar	3642	\LWR@itemizeitem	4704
\LWR@epubprintpendingfootnotes ..	3873	LWR@latestautopage (counter)	323
LWR@exitingtabular (boolean)	303	LWR@lateximagedepth (counter)	354
LWR@externalfilecnt (counter)	337	\LWR@lateximagedepthref	5729
\LWR@filenamenoblanks	3893	LWR@lateximagenumber (counter) ..	354
\LWR@filestart	4279	\LWR@lateximagenumberref	5732
\LWR@findword	379	\LWR@lateximagesfile	447
\LWR@floatbegin	5881	\LWR@lightrulewidth (length)	288
\LWR@floatend	5899	LWR@LIpage (counter)	354
\LWR@footnotetext	3808	\LWR@loadafter	111
\LWR@forceminwidth	7107	\LWR@loadbefore	122
\LWR@forcenewpage	3338	\LWR@loadnever	131
LWR@freeze thisfloat (boolean)	322	\LWR@longtabledatacaptiontag ...	5376
\LWR@future non spacelet	4779	\LWR@lookforpackagename	382
\LWR@FVstyle	17	\LWR@LwarpEnd	4398, 7495

\LWR@LwarpStart	4329, 7495	\LWR@printthetitle	4462
\LWR@maketitlesetup	33, 4503	\LWR@ProvidesPackageDrop	434
\LWR@maybeinthisfloat	5916	\LWR@ProvidesPackagePass	422
\LWR@maybenewtablerow	5088	\LWR@pushoneclose	4057
LWR@midrulecounter (counter)	288	\LWR@quickfile	444
LWR@midrules	288	\LWR@requesttoc	4391
LWR@minipagedepth (counter)	369	\LWR@requirepackagenames	378
LWR@minipagefullwidth (boolean) ..	371	\LWR@restoreorigaccents	617
LWR@minipageheight (length)	369	\LWR@restoreorigformatting	6270
\LWR@minipagestartpars	7339	\LWR@restoreorigprintxcolor	6926
\LWR@minipagestoppars	7342	\LWR@rotboxorigin	193
LWR@minipagethispar (boolean)	371	\LWR@rotstyle	91
LWR@minipagewidth (length)	369	\LWR@rule	7392
\LWR@multicolother	5289	\LWR@scalestyle	94
\LWR@multicolpartext	5284	\LWR@section	4067
\LWR@multicolskip	5295	\LWR@sectionnumber	4051
\LWR@multirowborder	3	\LWR@setexparray	3226
\LWR@mynexttoken	4778	\LWR@setlatestname	5701
\LWR@myshorttoc	5998	\LWR@setOSWindows	29
\LWR@nameref	5722	\LWR@sidetoc	6067
LWR@nestspan (environment)	3425	\LWR@singledollar	6216
\LWR@newhtmlfile	3959	LWR@skipatbang (boolean)	268
\LWR@newlabel	5759	LWR@skippingmcolrowcell (boolean) .	268
\LWR@newref	5809	LWR@skippingmrowcell (boolean) ...	268
LWR@nextautofloat (counter)	323	LWR@spandepth (counter)	210
LWR@nextautopage (counter)	323	\LWR@splabel	5735
LWR@nextequation (counter)	339	LWR@starredlongtable (boolean) ...	296
\LWR@nohspace	7380	LWR@startedrow (boolean)	268
\LWR@nullfonts	7236	\LWR@startnewdepth	4058
\LWR@nullifyNoAutoSpacing	5590	\LWR@startpars	3706
\LWR@openparagraph	3647	\LWR@startref	5772
\LWR@origcolspec	4921	\LWR@stoppars	3722
\LWR@originname	194	\LWR@stripperperiod	5700
\LWR@originnames	201	\LWR@strresult	4919
\LWR@parseaftercolumn	4981	\LWR@subaddcmidruletrim	5233
\LWR@parseatcolumn	4927	\LWR@subcmidrule	5218
\LWR@parsebangcolumn	4951	\LWR@subhtml elementclass	3534
\LWR@parsebarcolumn	4991	\LWR@subhyperref	5833
\LWR@parsebeforecolumn	4972	\LWR@subhyperrefclass	5837
\LWR@parseDcolumn	5018	\LWR@subinlineimage	5874
\LWR@parsenormalcolumn	5002	\LWR@subminipage	6995
\LWR@parsepcolumn	5014	\LWR@subnewref	5803
\LWR@parsetablecols	5023	\LWR@subsublabel	5738
\LWR@patchlists	4767	\LWR@subtableofcontents	6013
LWR@prevFileDepth (counter)	231	\LWR@syncmathjax	6250
\LWR@printatbang	5124	\LWR@tablecolspec	4918
\LWR@printbartag	5110	LWR@tablecolspos (counter)	275
\LWR@printmccoldata	5298	LWR@tablecolswidth (counter)	275
\LWR@printmccoltype	5268	\LWR@tabledatacolumn tag	5510
\LWR@printpendingfootnotes	3863	\LWR@tabledatasinglecolumn tag ..	5157

LWR@tableparcell (boolean)	268	MathJax summary	88, 336
LWR@tabletotalcols (counter)	275	mathsvg option	116
LWR@tabletotalcolsnext (counter)	275	showing as HTML	57, 107
LWR@tabular (environment)	5603	SVG summary	87, 335
\LWR@tabularendoffline	4895	math (environment)	6247
\LWR@tabularfinishrow	4870	MathJax	
\LWR@tabularhtmlcolumns	5442	mathjax option	116
\LWR@tabularleftedge	5149	subequations	88, 336
LWR@tabularmutemods (boolean)	303	summary	88, 336
LWR@tempheight (length)	386	tagged equations	88, 336
LWR@tempraise (length)	386	MathJax (program)	88, 336
LWR@tempwidth (length)	386	MathJax (program) [requirement]	45
\LWR@thiscmidrulewidth (length)	288	mathjax (boolean)	115
\LWR@thisfilename	3891	mathjax (option)	60, 70, 116
\LWR@thisfloat (counter)	322	mathsvg (option)	60, 70, 116
\LWR@thisnewfilename	3892	mcaption (package)	474
\LWR@titlingmaketitle	76, 4551	\mcolrowcell	5571
\LWR@topnavigation	3309	mdframed (package)	474
\LWR@traceinfo	506	\mdseries	7268
LWR@tracinglwarp (boolean)	133	memoir	
LWR@trimlrules	288	verse	563
LWR@trimrrules	288	memoir (package)	563
LWR@validtablecol (boolean)	280	meta tag, HTML	
LWR@verbtags (boolean)	258	author	74, 218
\LWR@vspace	7382	description	74, 218
LWRprint@fminipage (environment)	7164	generator	239
\LWRsetnextfloat	5924	viewport	240
\LyX	7476	metalo (package)	483
M			
Mac OS (program)	71, 114	microtype (package)	70, 124, 483
Madcap (program)	43	midfloat (package)	484
\mainmatter	4041	\MiKTeX	7475
\makebox	7079	minipage	
makeidx (package)	127	framed	79
\MakeIndex	7468	horizontal space between	384
\maketitle	40, 73, 4515	minipage (environment)	7016
margin		\minipagefullwidth	7008
numbers	111	misplaced \noalign	94, 266
tags	111	misplaced alignment tab character &	92, 264, 271
marginfit (package)	473	missing sections	61
marginfix (package)	473	moreverb (package)	484
marginnote (package)	473	mparhack (package)	486
\marginpar	3880	\mrowcell	5568
\markboth	3343	MS-Windows (program)	71, 114
\markright	3344	multicol (package)	486
markup languages	43	multicolumn	
math		with multirow	490
font size — SVG	87, 335, 354	\multicolumnrow	33, 5463
mathjax option	116	multirow	
		with multicolumn	490

<code>\multirow</code>	488	<code>lwarpmk</code>	116
<code>multirow</code> (package)	487	<code>mathjax</code>	60, 70, 116
<code>multline</code> (environment)	6469	<code>mathsvg</code>	60, 70, 116
<code>multline*</code> (environment)	6493	<code>OSWindows</code>	71, 114, 117
N			
<code>\Nameref</code>	5830	<code>warpHTML</code>	70, 115
<code>\nameref</code>	5821	<code>warpprint</code>	70, 115
<code>nameref</code> (package)	491	<code>xdyFilename</code>	76, 116
<code>needspace</code> (package)	491	<code>\OSPathSymbol</code>	28
<code>newclude</code> (package)	97, 491	<code>OSWindows</code> (option)	71, 114, 117
<code>newfloat</code>		P	
with <code>trivfloat</code> , algorithmic	560	packages	
<code>\newfloat</code>	2, 4	required	123
<code>newfloat</code> (package)	127	packages:	
<code>\newline</code>	7332	a4	395
<code>\newtheorem</code>	259	a4wide	395
<code>newtxmath</code> (package)	89	a5comb	395
<code>newunicodechar</code> (package)	69, 492	abstract	395
<code>nextpage</code> (package)	492	addlines	398
<code>nicefrac</code> (package)	89	adjmulticol	397
<code>\nolinebreak</code>	7385	afterpackage	125
<code>\nolinkurl</code>	5854	afterpage	398
<code>nonumonpart</code> (package)	492	algorithmicx	399
<code>\nopagebreak</code>	7389	alltt	399
<code>\nopagecolor</code>	44	amsmath	128
<code>nopageno</code> (package)	493	amsthm	400
<code>\normalfont</code>	7276	anonchap	402
<code>nowidow</code> (package)	493	anysize	403
<code>ntheorem</code> (package)	493	appendix	403
<code>\numberline</code>	6085	arabicfront	404
numbers		array	310
left margin	111	authblk	404
<code>numindex</code> (option) [<code>tocbibind</code>] ...	87, 552	babel	98
O			
<code>OpenOffice</code> (program)	42	balance	405
options:		bigdelim	405
[<code>lwarpmk</code>]:		bigstrut	407
<code>htmlglossary</code>	86, 446	bookmark	407
<code>printglossary</code>	86, 446	booktabs	407
[<code>tocbibind</code>]:		boxedminipage	408
<code>numindex</code>	87, 552	boxedminipage2e	408
[<code>tocloft</code>]:		calc	127
<code>titles</code>	84	caption	127
<code>BaseJobname</code>	116	caption2	409
<code>HomeHTMLFilename</code>	60, 117	ccaption	409
<code>HTMLFilename</code>	60, 117	change page	409
<code>IndexLanguage</code> ..	67, 86, 116, 446, 551	chappg	410
<code>latexmk</code>	60, 117	chn gpage	409
		cleveref	366
		cmap	70
		color	410

comment	118	hyperref	317, 457
crop	410	hyperxmp	462
cuted	411	idxlayout	462
cutwin	411	ifplatform	113
dblfnote	412	indentfirst	463
dcolumn	412	inputenc	69, 463
draftwatermark	412	keyfloat	463
ebook	413	kvoptions	115
ellipsis	413	layout	464
emptypage	414	letltxmacro	113
endnotes	414	letterspace	465
enumerate	415	lettrine	465
enumitem	415	lips	466
environ	127	listings	467
epigraph	416	lmodern	69
eso-pic	417	longtable	470
etoolbox	113	lscape	472
everyhook	126	ltxcaption	472
everypage	417	lwarp	70
everyshi	418	makeidx	127
expl3	126	marginfit	473
extramarks	418	marginfix	473
fancybox	419	marginnote	473
fancyhdr	422	mcaption	474
fancyvrb	423	mdframed	474
figcaps	428	memoir	563
filecontents	126	metalogo	483
fix2col	429	microtype	70, 124, 483
float	429	midfloat	484
floatflt	430	moreverb	484
floatpag	431	mparhack	486
floatrow	431	multicol	486
flushend	437	multirow	487
fncychap	437	nameref	491
fnpos	438	needspace	491
fontenc	69, 438	newclude	97, 491
fontspec	69, 439	newfloat	127
footmisc	439	newtxmath	89
footnote	440	newunicodechar	69, 492
footnotehyper	441	nextpage	492
framed	441	nicefrac	89
ftnright	444	nonumonpart	492
fullpage	444	nopageno	493
fullwidth	444	nowidow	493
geometry	125, 445	ntheorem	493
getttitlestring	126	pagenote	504
glossaries	85, 445	paralist	504
graphics	361, 447	parskip	505
graphicx	361, 457	pdflscape	505
grffile	457	pdfsync	505

pfnote	506	wallpaper	564
placeins	506	wrapfig	565
prelim2e	506	xcolor	566
quotchap	507	xfrac	574
ragged2e	508	xifthen	126
realscripts	509	xltextra	576
refcount	127	xmpincl	577
relsize	510	xparse	125
romanbar	511	xstring	126, 127
romanbarpagenumber	511	xtab	577
rotating	512	zref	127
rotfloat	512	zwpagelayout	579
savetrees	513	page	
scaleft	513	inaccessible	61
sectsty	513	\pagebreak	7386
setspace	514	\pagecolor	43
shadow	515	pagenote (package)	504
showidx	516	\pagenumbering	3349
showkeys	516	\pageref	5816
sidecap	516	\pagerefPageFor	5815
sidenotes	517	\pagestyle	3341
siunitx	89, 361, 519	Pandoc (program)	42
soul	519	\paragraph	4254
stabular	521	paralist (package)	504
subfig	521	\parbox	7072
supertabular	527	\parsemulticolumnalignment	5325
syntonly	528	parskip (package)	505
tabls	529	\part	4205
tabularx	529	pdfcrop (program) [requirement]	45
tabulary	530	pdfLaTeX (program) [requirement]	45
textarea	530	pdfscape (package)	505
textcomp	70, 531	pdfseparate (program) [requirement]	45, 49
textpos	534	pdfsync (package)	505
theorem	534	pdftocairo (program) [requirement]	45, 49
threeparttable	538	pdftotext (program) [requirement]	45, 49
tikz	539	Perl	50
titleps	540	perl (program) [requirement]	50
titleref	543	pfnote (package)	506
titlesec	543	\phantomsection	7437
titletoc	545	picture (environment)	368, 6971
titling	546	placeins (package)	506
tocbibind	87, 551	Plastex (program)	41
tocloft	553	\popclose	3199
transparent	558	Poppler	45, 49
trivfloat	559	prelim2e (package)	506
typearea	560	\printauthor	246, 4470, 4489
ulem	561	\printdate	246, 4481, 4491
units	89	printglossary (option) [lwarpmk]	86, 446
upref	562	\printindex	6135
verse	562, 563		

\PrintStack	3317	\pushclose	3172
\printthanks	246, 248		
\printtitle	246, 4454, 4488		
problems	106		
programs			
utility	44		
programs:			
[requirement]:			
LuaLaTeX	45		
MathJax	45		
pdfcrop	45		
pdfLaTeX	45		
pdfseparate	45, 49		
pdftocairo	45, 49		
pdftotext	45, 49		
perl	50		
XeLaTeX	45		
Adobe	43		
AsciiDoc	42		
AsciiDoctor	42		
Asciidoctor-LaTeX	42		
Flare	43		
FrameMaker	43		
GELLMU	41		
GladTeX	42		
Hevea	41		
InDesign	43		
LaTeX2HTML	41		
LaTeXML	41		
LibreOffice	42		
Linux	71, 114		
lwarpmk	104, 179		
Mac OS	71, 114		
Madcap	43		
MathJax	88, 336		
MS-Windows	71, 114		
OpenOffice	42		
Pandoc	42		
Plastex	41		
TeX2page	41		
TeX4ht	41		
TtH	41		
Unix	71, 114		
Windows	71, 114		
Word	42		
xindy	75		
project.css (file)	75		
project.lwarpmkconf (file)	139		
project_html.tex (file)	138		
published	253		
		Q	
		\qqquad	385, 7350
		\quad	385, 7345
		quotchap (package)	507
		quote (environment)	4622
		R	
		ragged2e (package)	508
		\raggedbottom	3345
		\raggedleft	242
		\raggedright	242
		\raisebox	7157
		realscripts (package)	509
		\ref	5809
		refcount (package)	127
		\reflectbox	247
		relsize (package)	510
		[requirement]:	
		LuaLaTeX (program)	45
		MathJax (program)	45
		pdfcrop (program)	45
		pdfLaTeX (program)	45
		pdfseparate (program)	45, 49
		pdftocairo (program)	45, 49
		pdftotext (program)	45, 49
		perl (program)	50
		XeLaTeX (program)	45
		\RequirePackage	398
		\resizebox	252
		\restylefloat	17
		\ResumeTabular	5497
		\rmfamily	7270
		romanbar (package)	511
		romanbarpagenumber (package)	511
		\rotatebox	213
		rotating (package)	512
		rotfloat (package)	512
		\rule	7434
		S	
		sample_project.css (file)	75, 176
		savetrees (package)	513
		\sb	7278
		\scalebox	232
		scalefnt (package)	513
		\scshape	7275
		section	
		depths	109

missing	61	tables (package)	529
\section	4229	tabular	
sectsty (package)	513	baseline	280
\SetHTMLFileNumber	3255	HTML columnn conversion	281
setspace (package)	514	macros inside	93, 266
settings		misplaced \noalign	94, 266
css project-specific	75	misplaced alignment tab character &	92, 264
CSS selection	75	multicolumn with multirow	490
HTML conversion	60	row corruption	93, 266
lwrap package options	70	\TabularMacro	5495
selecting output	72	tabularx (package)	529
title page	73	tabulary (package)	530
\sffamily	7271	tagged equations	
\sfrac	574	Mathjax	88, 336
shadow (package)	515	\TeX	7442
showidx (package)	516	TeX2page (program)	41
showkeys (package)	516	TeX4ht (program)	41
sidecap (package)	516	textarea (package)	530
sidenotes (package)	517	\textbf	7198
SideTOCDepth (counter)	61, 329	\textcolor	33
\sidetocname	6064	textcomp (package)	70, 531
\simplechapterdelim	4065	\textgreater	3245
siunitx (package)	89, 361, 519	\textit	7213
\sloppy	3347	\textless	3245
soul (package)	519	\textmd	7195
\sp	7277	\textnormal	7222
space		textpos (package)	534
horizontal	76	\textrm	7201
between minipages	384	\textsc	7216
split		\textsf	7204
miss-numbered	87, 89, 335, 494	\textsl	7219
stabular (package)	521	\textsubscript	7280
stack depths	109	\textsuperscript	7279
\StartDefiningTabulars	4849	\texttt	7207
subequations		\textup	7210
MathJax	88, 336	\tfl@chapter@fix	559
subfig (package)	521	\thanks	74
\subparagraph	4262	\thanksmarkseries	84
\subsection	4237	\theauthor	246
\subsubsection	4245	\thedate	246
subtitle	253	theindex (environment)	6142
supertabular (package)	527	theorem (package)	534
SVG		\thetitle	246
images showing as HTML	57, 107	\thispagestyle	3342
math summary	87, 335	threeparttable (package)	538
mathsvg option	116	tikz	
syntonly (package)	528	catcodes	539
		dollar redefined	539
T		tikz (package)	539
\tableofcontents	63		

tikzpicture (environment)	8	file & section names	228
\title	73	index	70
titlepage		selection	69
subtitle and published	253	utility	
titlepage (environment)	73, 4438	programs	44
titles (package)	540		
titleref (package)	543	V	
titles (option) [tocloft]	84	Verbatim (environment)	13
titlesec (package)	543	verbatim (environment)	4679
titletoc (package)	545	\VerbatimHTMLWidth (length)	257
titling (package)	546	\verbatiminput	4671
titlingpage (environment)	14, 73	verse (environment)	2
[tocbibind]:		verse (package)	562, 563
numindex (option)	87, 552	viewport	
tocbibind (package)	87, 551	HTML meta tag	240
tocdepth (counter)	60	vleftmargini (length)	97, 256, 563
[tocloft]:		vleftskip (length)	97, 256, 563
titles (option)	84	\vspace	7383
tocloft (package)	553		
\tracinglwarp	108, 505	W	
transparent (package)	558	wallpaper (package)	564
trivfloat		warning icon	111
with newfloat, algorithmic	560	warpall (environment)	72, 100
trivfloat (package)	559	warpHTML (environment)	50, 64, 72, 101
troubleshooting	106	warpHTML (option)	70, 115
HTML debug comments	133	\warpHTMLonly	64, 72, 90
\ttfamily	7272	warpingHTML (boolean)	115
TtH (program)	41	warpingprint (boolean)	115
tutorial.tex (file)	51	warpprint (environment)	63, 72, 101
typearea (package)	560	warpprint (option)	70, 115
		\warpprintonly	64, 72, 89
U		Windows (program)	71, 114
ulem (package)	561	Word (program)	42
\underline	7283	word processor	
underscore		HTML conversion settings	101, 102, 134
filename	60, 77	wrapfig (package)	565
Unicode			
enhanced coverage	69	X	
file & section names	228	xcolor (package)	566
input characters	112	xdyFilename (option)	76, 116
selection	69	XeLaTeX	
units (package)	89	detection	124
Unix (program)	71, 114	file & section names	228
\up	7281	\XeLaTeX	7461
upref (package)	562	XeLaTeX (program) [requirement]	45
\upshape	7273	\XeTeX	7461
\url	5861	xfrac (package)	574
usingOSWindows (boolean)	114	\xfracHTMLfontsize	3
UTF-8		xifthen (package)	122
enhanced coverage	69		

xindy		xstring (package)	126, 127
customizing	75	xtab (package)	577
xindy (program)	75		
xltxtra (package)	576		
xmpincl (package)	577		
xparse (package)	125		
		Z	
		zref (package)	127
		zwpagelayout (package)	579