



The **lwarp** package

ℒ_Ḓ to HTML

v0.44 — 2017/11/22

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

Documents may be produced by pdfℒ_Ḓ, Luaℒ_Ḓ, or Xeℒ_Ḓ. 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. Almost two hundred ℒ_Ḓ packages are supported with these high-level source compatibility replacements, and many others work as-is.

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

[To update existing projects, see section 2, Updates.](#)

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

License:

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

1 Supporting T_EX development

T_EX development:

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

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

Donations may be directed towards individual projects:

TUG Bursary Fund: Assistance for attending annual TUG meetings.

CTAN: The Comprehensive T_EX Archive Network — Central storage for T_EX.

TeX Development Fund: Support for specific projects.

EduTeX: Teaching and using T_EX in schools and universities.

GUST e-foundry fonts: Enhanced for math and additional language groups.

LaTeX Project: Modernizing the L^AT_EX core.

Libre Font Fund: Fonts, tools (FontForge), and distribution (the Open Font Library).

LuaTeX: Combining the pdfT_EX engine and the Lua language.

MetaPost: Postscript graphics.

MacTeX: T_EX for Mac.

PDF Accessibility: Modern PDF standards.

Other: Additional projects may be specified.

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

For users of MiK_TE_X: <https://miktex.org/donations.html>

For country-specific T_EX users groups: <http://tug.org/usergroups.html>

Contents

1	Supporting TeX development	2
	List of Figures	33
	List of Tables	33
2	Updates	34
3	Introduction	42
3.1	Supported packages and features	43
4	Alternatives	47
4.1	Internet class	47
4.2	TeX4ht	47
4.3	Translators	47
4.4	AsciiDoc	48
4.5	Pandoc	48
4.6	Word processors	48
4.7	Commercial systems	49
4.8	Comparisons	49
5	Installation	50
5.1	Installing the lwarp package	50
5.2	Installing the lwarpmk utility	53
5.2.1	Using a local copy of lwarpmk	54
5.3	Installing additional utilities	55
6	Tutorial	57
6.1	Starting a new project	57
6.2	Compiling the print version with lwarpmk	61
6.3	Compiling the HTML version with lwarpmk	62
6.4	Generating the SVG images	63
6.5	Using MathJax for math	64
6.6	Changing the CSS style	65
6.7	Customizing the HTML output	65
6.8	Using latexmk	66
6.9	Using XeLaTeX or LuaLaTeX	67
6.10	Using a glossary	68
6.11	Cleaning auxiliary files	69
6.12	Cleaning auxiliary and output files	69
6.13	Processing multiple projects in the same directory	69
6.14	Using the make utility	69
7	Additional details	70
7.1	Font and UTF-8 support	70

7.1.1	Indexes and UTF-8	71
7.2	lwarp package loading and options	72
7.3	Customizing the HTML output	74
7.3.1	Example HTML file naming	78
7.4	Customizing the CSS	79
7.5	Selecting the operating system	80
7.6	Selecting actions for print or HTML output	81
7.7	Commands to be placed into the warpprint environment	82
7.8	Title page	82
7.9	HTML page meta descriptions	83
7.10	HTML page meta title	84
7.11	HTML page meta author	84
7.12	Modifying xindy index processing	84
8	Special cases and limitations	85
8.1	Formatting	85
8.1.1	Text formatting	85
8.1.2	Horizontal space	85
8.1.3	Text alignment	85
8.1.4	Accents	85
8.1.5	Textcomp	86
8.1.6	Superscripts and other non-math uses of math mode	86
8.1.7	Empty <code>\item</code> followed by a new line of text or a nested list:	86
8.1.8	Filenames and URLs in lists or footnotes	86
8.1.9	resize package	86
8.2	Boxes and minipages	86
8.2.1	Marginpars	86
8.2.2	Save Boxes	87
8.2.3	Minipages	87
8.2.4	Side-by-side minipages	88
8.2.5	Framed minipages and other environments	88
8.2.6	fancybox package	89
8.2.7	mdframed package	90
8.3	Cross-references	91
8.3.1	Page references	91
8.3.2	cleveref and varioref packages	92
8.3.3	Hyperlinks, hyperref, and url	92
8.3.4	Footnotes and page notes	92
8.4	Front and back matter	92
8.4.1	Starred chapters and sections	92
8.4.2	abstract package	93
8.4.3	titling and authblk	93
8.4.4	tocloft package	94
8.4.5	appendix package	94
8.4.6	pagenote package	94
8.4.7	endnotes package	94

8.4.8	glossaries package	94
8.4.9	Index and the tocbibind package	95
8.5	Math	96
8.5.1	Rendering tradeoffs	96
8.5.2	SVG option	97
8.5.3	MathJax option	97
8.5.4	ntheorem package	99
8.5.5	siunitx package	99
8.5.6	units and nicefrac packages	99
8.5.7	newtxmath package	100
8.6	Graphics	100
8.6.1	grffile package	101
8.6.2	color package	101
8.6.3	xcolor package	101
8.6.4	overpic package	102
8.7	Tabular	102
8.7.1	longtable package	105
8.7.2	supertabular and xtab packages	105
8.7.3	bigdelim package	105
8.8	Floats	106
8.8.1	float, trivfloat, and/or algorithmicx together	106
8.8.2	caption and subcaption packages	106
8.8.3	subfig package	106
8.8.4	floatrow package	106
8.8.5	keyfloat package	107
8.9	Koma-Script	107
8.10	Miscellaneous	107
8.10.1	verse and memoir	107
8.10.2	newclude package	108
8.10.3	babel package	108
8.10.4	todonotes and luatodonotes packages	109
8.10.5	fixme	109
9	EPUB conversion	110
10	Word-processor conversion	112
10.1	Activating word-processor conversion	112
10.2	Additional modifications	114
10.3	Recommendations	115
10.4	Limitations	116
11	Modifying lwarp	118
11.1	Creating an lwarp version of a package	118
11.2	Adding a package to the lwarp.dtx file	119
11.3	Testing lwarp	119
11.4	Modifying lwarpmk	119

12	Troubleshooting	121
12.1	Using the <code>lwarp.sty</code> package	121
12.1.1	Debug tracing output	123
12.2	Compiling the <code>lwarp.dtx</code> file	123
1	lwarp.sty	124
13	Implementation	124
14	Section depths and HTML headings	125
15	Source Code	126
16	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	127
17	pdf_lat_EX T1 and UTF8 encoding	127
18	Unicode input characters	127
19	Early package requirements	129
20	Operating-System portability	130
20.1	Common portability code	130
20.2	Unix, Linux, and Mac OS	130
20.3	MS-Windows	130
21	Package options	131
21.1	Conditional compilation	134
22	Misplaced packages	135
23	Required packages	140
24	Loading packages	145
25	File handles	147
26	Include a file	148
27	Copying a file	149
28	Debugging messages	150
29	Koma-script	150
30	HTML-conversion output modifications	151
30.1	User-level controls	151
30.2	Heading adjustments	153

31	Remembering original formatting macros	154
32	Accents	156
33	Configuration Files	158
33.1	project_html.tex	158
33.2	lwarpmk.conf	158
33.3	project.lwarpmkconf	159
33.4	lwarp.css	160
33.5	lwarp_sagebrush.css	187
33.6	lwarp_formal.css	192
33.7	sample_project.css	196
33.8	lwarp.xdy	197
33.9	lwarp_mathjax.txt	197
33.10	lwarpmk option	199
34	Stacks	211
34.1	Assigning depths	211
34.2	Closing actions	212
34.3	Closing depths	212
34.4	Pushing and popping the stack	213
35	Data arrays	214
36	HTML entities	215
37	HTML filename generation	216
38	Homepage link	218
39	\PrintStack diagnostic tool	219
40	Closing stack levels	219
41	PDF pages and styles	220
42	HTML tags, spans, divs, elements	221
42.1	Mapping \LaTeX Sections to HTML Sections	221
42.2	Babel-French	222
42.3	HTML tags	223
42.4	Block tags and comments	225
42.5	Div class and element class	226
42.6	Single-line elements	227
42.7	HTML5 semantic elements	228
42.8	High-level block and inline classes	228
42.9	Closing HTML tags	230
43	Paragraph handling	231

44	Paragraph start/stop handling	234
45	Page headers and footers	237
46	CSS	238
47	Title, HTML meta author, HTML meta description	239
48	Footnotes	240
48.1	Regular page footnotes	240
48.2	Minipage footnotes	240
48.3	Titlepage thanks	240
48.4	Regular page footnote implementation	241
48.5	Minipage footnote implementation	243
48.6	Printing pending footnotes	244
49	Marginpars	244
50	Splitting HTML files	246
51	Sectioning	251
51.1	User-level starred section commands	251
51.2	Book class commands	252
51.3	Sectioning support macros	253
51.4	\section and friends	259
52	Starting a new file	261
53	Starting HTML output	264
54	Ending HTML output	267
55	Title page	268
55.1	Setting the title, etc.	269
55.2	Changes for \affiliation	270
55.3	Printing the thanks	271
55.4	Printing the title, etc. in HTML	272
55.5	Printing the title, etc. in print form	273
55.6	\maketitle for HTML output	273
55.7	\published and \subtitle	276
56	Abstract	278
57	Quote and verse	278
57.1	Citations and attributions	278
57.2	Quotes, quotations	279
57.3	Verse	279

58	Verbatim and tabbing	281
59	Theorems	283
60	Lists	284
60.1	List environment	285
60.2	Itemize	286
60.3	Enumerate	286
60.4	Description	287
60.5	Patching the lists	288
61	Tabular	289
61.1	Limitations	289
61.2	Token lookahead	292
61.3	Booleans	293
61.4	Handling &, @, !, and bar	294
61.4.1	Localizing & catcodes	296
61.4.2	Handling &	296
61.4.3	Filling an unfinished row	297
61.5	Handling \	299
61.6	Variables	300
61.7	Parsing @, >, <, !, bar columns	301
61.8	Parsing 'l', 'c', or 'r' columns	304
61.9	Parsing 'p', 'm', or 'b' columns	305
61.10	Parsing 'D' columns	305
61.11	Parsing the column specifications	305
61.12	Starting a new row	309
61.13	Printing vertical bar tags	310
61.14	Printing at or bang tags	310
61.15	Data opening tag	312
61.16	Midrules	313
61.17	Multicolumns	318
61.17.1	Parsing multicolumns	318
61.17.2	Multicolumn factored code	321
61.17.3	Multicolumn	323
61.17.4	Longtable captions	324
61.17.5	Counting HTML tabular columns	326
61.18	Multicolumnrow	328
61.19	Utility macros inside a table	329
61.20	Special-case tabular markers	330
61.21	Checking for a new table cell	331
61.22	\mrowcell	333
61.23	\mcolrowcell	333
61.24	New \tabular definition	334
61.25	Array	339

62	Cross-references	339
62.1	Setup	339
62.2	Zref setup	341
62.3	Labels	342
62.4	References	344
62.5	Hyper-references	346
63	Floats	349
63.1	Float captions	350
63.1.1	Caption inside a float environment	352
63.1.2	Caption and LOF linking and tracking	352
64	Table of Contents, LOF, LOT	355
64.1	Reading and printing the TOC	356
64.2	High-level TOC commands	359
64.3	Side TOC	359
64.4	Low-level TOC line formatting	360
65	Index and glossary	363
66	Restoring original formatting	365
67	Math	367
67.1	Limitations	367
67.1.1	Rendering tradeoffs	367
67.1.2	SVG option	368
67.1.3	MathJax option	368
67.2	Inline and display math	370
67.3	MathJax support	373
67.4	Equation environment	375
67.5	AMS Math environments	377
67.5.1	Support macros	377
67.5.2	Environment patches	379
68	Lateximages	385
68.1	Description	385
68.2	Support counters and macros	386
68.3	Font size	387
68.4	Sanitizing math expressions for HTML	387
68.5	Equation numbers	388
68.6	HTML <alt> tags	390
68.7	lateximage	390
69	center, flushleft, flushright	394
70	Pre-loaded packages	395

71	Siunitx	396
72	Graphics print-mode modifications	396
72.1	General limitations	396
72.2	Print-mode modifications	397
73	Xcolor boxes	398
74	Cleveref	401
75	Picture	403
76	Boxes and Minipages	404
76.1	Counters and lengths	404
76.2	Footnote handling	405
76.3	Minipage handling	405
76.4	Parbox, makebox, framebox, fbox, raisebox	409
77	Direct formatting	414
78	Skips, spaces, font sizes	420
79	\phantomsection	428
80	\LaTeX and other logos	428
80.1	HTML logos	429
80.2	Print logos	431
81	\AtBeginDocument, \AtEndDocument	431
82	Trademarks	432
2	lwarp-patch-komascript.sty	433
83	patch-komascript	433
3	lwarp-a4.sty	434
84	a4	434
4	lwarp-a4wide.sty	435
85	a4wide	435

5	lwarp-a5comb.sty	435
86	a5comb	435
6	lwarp-abstract.sty	435
87	abstract	435
7	lwarp-adjmulticol.sty	437
88	adjmulticol	437
8	lwarp-addlines.sty	438
89	addlines	438
9	lwarp-afterpage.sty	438
90	afterpage	438
10	lwarp-algorithmicx.sty	439
91	algorithmicx	439
11	lwarp-alltt.sty	440
92	alltt	440
12	lwarp-amsthm.sty	440
93	amsthm	440
13	lwarp-anonchap.sty	443
94	anonchap	443
14	lwarp-anysize.sty	443
95	anysize	443

15	lwarp-appendix.sty	444
96	appendix	444
16	lwarp-arabicfront.sty	444
97	arabicfront	444
17	lwarp-atbegshi.sty	445
98	atbegshi	445
18	lwarp-authblk.sty	445
99	authblk	445
19	lwarp-balance.sty	446
100	balance	446
20	lwarp-bigdelim.sty	446
101	bigdelim	446
21	lwarp-bigstrut.sty	448
102	bigstrut	448
22	lwarp-bookmark.sty	448
103	bookmark	448
23	lwarp-booktabs.sty	449
104	booktabs	449
24	lwarp-boxedminipage.sty	449
105	boxedminipage	449

25	lwarp-boxedminipage2e.sty	449
106	boxedminipage2e	449
26	lwarp-breakurl.sty	450
107	breakurl	450
27	lwarp-cancel.sty	450
108	cancel	450
28	lwarp-caption2.sty	451
109	caption2	451
29	lwarp-ccaption.sty	452
110	ccaption	452
30	lwarp-changepage.sty	452
111	changepage	452
31	lwarp-chngpage.sty	452
112	chngpage	452
32	lwarp-chappg.sty	453
113	chappg	453
33	lwarp-color.sty	453
114	color	453
34	lwarp-crop.sty	453
115	crop	453

35	lwarp-cuted.sty	454
116	cuted	454
36	lwarp-cutwin.sty	454
117	cutwin	454
37	lwarp-dblfnote.sty	455
118	dblfnote	455
38	lwarp-dcolumn.sty	455
119	dcolumn	455
39	lwarp-draftwatermark.sty	456
120	draftwatermark	456
40	lwarp-easy-todo.sty	456
121	easy-todo	456
41	lwarp-ebook.sty	457
122	ebook	457
42	lwarp-ellipsis.sty	457
123	ellipsis	457
43	lwarp-emptypage.sty	458
124	emptypage	458
44	lwarp-endnotes.sty	458
125	endnotes	458

45	lwarp-enumerate.sty	459
126	enumerate	459
46	lwarp-enumitem.sty	459
127	enumitem	459
47	lwarp-epigraph.sty	460
128	epigraph	460
48	lwarp-eso-pic.sty	461
129	eso-pic	461
49	lwarp-everypage.sty	461
130	everypage	461
50	lwarp-everyshi.sty	462
131	everyshi	462
51	lwarp-extramarks.sty	462
132	extramarks	462
52	lwarp-fancybox.sty	463
133	fancybox	463
53	lwarp-fancyhdr.sty	467
134	fancyhdr	467
54	lwarp-fancyref.sty	467
135	fancyref	467

55	lwarp-fancyvrb.sty	469
136	fancyvrb	469
56	lwarp-figcaps.sty	474
137	figcaps	474
57	lwarp-fix2col.sty	474
138	fix2col	474
58	lwarp-fixme.sty	474
139	fixme	474
59	lwarp-fixmetodonotes.sty	476
140	fixmetodonotes	476
60	lwarp-float.sty	476
141	float and \newfloat	476
61	lwarp-floatflt.sty	478
142	floatflt	478
62	lwarp-floatpag.sty	479
143	floatpag	479
63	lwarp-floatrow.sty	479
144	floatrow	479
64	lwarp-flushend.sty	485
145	flushend	485

65	lwarp-fncychap.sty	485
146	fncychap	485
66	lwarp-fnpos.sty	486
147	fnpos	486
67	lwarp-fontenc.sty	486
148	fontenc	486
68	lwarp-fontspec.sty	487
149	fontspec	487
69	lwarp-footmisc.sty	487
150	footmisc	487
70	lwarp-footnote.sty	488
151	footnote	488
71	lwarp-footnotehyper.sty	489
152	footnotehyper	489
72	lwarp-framed.sty	489
153	framed	489
73	lwarp-ftnright.sty	492
154	ftnright	492
74	lwarp-fullpage.sty	492
155	fullpage	492

75	lwarp-fullwidth.sty	493
156	fullwidth	493
76	lwarp-geometry.sty	493
157	geometry	493
77	lwarp-glossaries.sty	494
158	glossaries	494
78	lwarp-graphics.sty	495
159	graphics	495
159.1	Graphics extensions	495
159.2	Length conversions and graphics options	496
159.3	Printing HTML styles	498
159.4	\includegraphics	499
159.5	Boxes	503
79	lwarp-graphicx.sty	506
160	graphicx	506
80	lwarp-grffile.sty	506
161	grffile	506
81	lwarp-hang.sty	507
162	hang	507
82	lwarp-hyperref.sty	508
163	hyperref	508

83	lwarp-hyperxmp.sty	513
164	hyperxmp	513
84	lwarp-idxlayout.sty	513
165	idxlayout	513
85	lwarp-ifoddpag.sty	514
166	ifoddpag	514
86	lwarp-indentfirst.sty	514
167	indentfirst	514
87	lwarp-inputenc.sty	515
168	inputenc	515
88	lwarp-keyfloat.sty	515
169	keyfloat	515
89	lwarp-layout.sty	516
170	layout	516
90	lwarp-letterspace.sty	517
171	letterspace	517
91	lwarp-lettrine.sty	517
172	lettrine	517

92	lwarp-lips.sty	518
173	lips	518
93	lwarp-listings.sty	519
174	listings	519
94	lwarp-longtable.sty	522
175	longtable	522
95	lwarp-lscape.sty	524
176	lscape	524
96	lwarp-ltcaption.sty	524
177	ltcaption	524
97	lwarp-ltxtable.sty	525
178	ltxtable	525
98	lwarp-luatodonotes.sty	525
179	luatodonotes	525
99	lwarp-marginfit.sty	527
180	marginfit	527
100	lwarp-marginfix.sty	527
181	marginfix	527

101	lwarp-marginnote.sty	528
182	marginnote	528
102	lwarp-mcaption.sty	528
183	mcaption	528
103	lwarp-mdframed.sty	529
184	mdframed	529
184.1	Package loading	529
184.2	Limitations	529
184.3	Color and length HTML conversion	530
184.4	Environment encapsulation	531
184.5	Titles and subtitles	532
184.6	New environments	535
104	lwarp-metalogo.sty	538
185	metalogo	538
105	lwarp-microtype.sty	538
186	microtype	538
106	lwarp-midfloat.sty	539
187	midfloat	539
107	lwarp-moreverb.sty	539
188	moreverb	539
108	lwarp-morewrites.sty	541
189	morewrites	541

109	lwarp-mparhack.sty	541
190	mparhack	541
110	lwarp-multicol.sty	541
191	multicol	541
111	lwarp-multirow.sty	543
192	multirow	543
192.1	Multirow	543
192.2	Combined multicolumn and multirow	545
112	lwarp-nameref.sty	546
193	nameref	546
113	lwarp-needspace.sty	546
194	needspace	546
114	lwarp-newclude.sty	546
195	newclude	546
115	lwarp-newunicodechar.sty	547
196	newunicodechar	547
116	lwarp-nextpage.sty	547
197	nextpage	547
117	lwarp-nonumonpart.sty	547
198	nonumonpart	547

118	lwarp-nopageno.sty	548
199	nopageno	548
119	lwarp-nowidow.sty	548
200	nowidow	548
120	lwarp-ntheorem.sty	548
201	ntheorem	548
201.1	Limitations	549
201.2	Options	549
201.3	Remembering the theorem style	550
201.4	HTML cross-referencing	551
201.5	\newtheoremstyle	552
201.6	Standard styles	552
201.7	Additional objects	554
201.8	Renewed standard configuration	555
201.9	amsthm option	555
201.10	Ending a theorem	559
201.11	\NoEndMark	559
201.12	List-of	559
201.13	Symbols	560
201.14	Cross-referencing	560
121	lwarp-overpic.sty	560
202	overpic	560
122	lwarp-pagenote.sty	561
203	pagenote	561
123	lwarp-paralist.sty	561
204	paralist	561
124	lwarp-parskip.sty	562
205	parskip	562

125	lwarp-pdfscape.sty	562
206	pdfscape	562
126	lwarp-pdfsync.sty	562
207	pdfsync	562
127	lwarp-pfnote.sty	563
208	pfnote	563
128	lwarp-placeins.sty	563
209	placeins	563
129	lwarp-prelim2e.sty	563
210	prelim2e	563
130	lwarp-preview.sty	564
211	preview	564
131	lwarp-quotchap.sty	564
212	quotchap	564
132	lwarp-ragged2e.sty	565
213	ragged2e	565
133	lwarp-realscripts.sty	566
214	realscripts	566

134	lwarp-reysize.sty	567
215	reysize	567
135	lwarp-romanbar.sty	568
216	romanbar	568
136	lwarp-romanbarpagenumber.sty	569
217	romanbarpagenumber	569
137	lwarp-rotating.sty	569
218	rotating	569
138	lwarp-rotfloat.sty	569
219	rotfloat	569
139	lwarp-savetrees.sty	570
220	savetrees	570
140	lwarp-scalefnt.sty	571
221	scalefnt	571
141	lwarp-scrextend.sty	571
222	scrextend	571
142	lwarp-scrhack.sty	574
223	scrhack	574
143	lwarp-sclayer.sty	574
224	sclayer	574

144	lwarp-sclayer-notecolumn.sty	576
225	sclayer-notecolumn	576
145	lwarp-sclayer-scrpage.sty	576
226	sclayer-scrpage	576
146	lwarp-section.sty	577
227	section	577
147	lwarp-sectsty.sty	578
228	sectsty	578
148	lwarp-setspace.sty	579
229	setspace	579
149	lwarp-shadow.sty	580
230	shadow	580
150	lwarp-showidx.sty	581
231	showidx	581
151	lwarp-showkeys.sty	581
232	showkeys	581
152	lwarp-sidecap.sty	581
233	sidecap	581
153	lwarp-sidenotes.sty	582
234	sidenotes	582

154	lwarp-siunitx.sty	584
235	siunitx	584
155	lwarp-soul.sty	585
236	soul	585
156	lwarp-soulpos.sty	587
237	soulpos	587
157	lwarp-soulutf8.sty	588
238	soulutf8	588
158	lwarp-stabular.sty	588
239	stabular	588
159	lwarp-subfig.sty	589
240	subfig	589
160	lwarp-supertabular.sty	594
241	supertabular	594
161	lwarp-syntonly.sty	596
242	syntonly	596
162	lwarp-tables.sty	596
243	tables	596
163	lwarp-tabularx.sty	597
244	tabularx	597

164	lwarp-tabulary.sty	597
245	tabulary	597
165	lwarp-textarea.sty	598
246	textarea	598
166	lwarp-textcomp.sty	598
247	textcomp	598
247.1	Limitations	598
247.2	Package loading	599
247.3	Remembering original defintions	599
247.4	HTML symbols	600
247.4.1	pdf \TeX symbols	600
247.4.2	X \TeX and Lua \TeX symbols	600
247.5	HTML dicritics	601
247.6	Inside a lateximage	601
167	lwarp-textpos.sty	602
248	textpos	602
168	lwarp-theorem.sty	603
249	theorem	603
249.1	Remembering the theorem style	603
249.2	CSS patches	605
169	lwarp-threeparttable.sty	607
250	threeparttable	607
170	lwarp-tikz.sty	608
251	tikz	608

171	lwarp-titleps.sty	609
252	titleps	609
172	lwarp-titleref.sty	612
253	titleref	612
173	lwarp-titlesec.sty	612
254	titlesec	612
174	lwarp-titletoc.sty	614
255	titletoc	614
175	lwarp-titling.sty	616
256	titling	616
176	lwarp-tocbasic.sty	621
257	tocbasic	621
177	lwarp-tocbibind.sty	621
258	tocbibind	621
178	lwarp-tocloft.sty	623
259	tocloft	623
179	lwarp-tocstyle.sty	630
260	tocstyle	630
180	lwarp-todo.sty	630
261	todo	630

181	lwarp-todonotes.sty	631
262	todonotes	631
182	lwarp-transparent.sty	633
263	transparent	633
183	lwarp-trivfloat.sty	633
264	trivfloat	633
264.1	Combining \newfloat, \trivfloat, and algorithmicx	634
184	lwarp-typearea.sty	635
265	typearea	635
185	lwarp-ulem.sty	635
266	ulem	635
186	lwarp-upref.sty	637
267	upref	637
187	lwarp-verse.sty	638
268	verse	638
188	lwarp-wallpaper.sty	640
269	wallpaper	640
189	lwarp-watermark.sty	640
270	watermark	640

190	lwarp-wrapfig.sty	641
271	wrapfig	641
191	lwarp-xcolor.sty	642
272	xcolor	642
272.1	Limitations	642
272.2	Xcolor definitions: location and timing	643
272.3	Package loading	645
272.4	Remembering and restoring original definitions	645
272.5	HTML color style	645
272.6	HTML border	647
272.7	High-level macros	647
192	lwarp-xfrac.sty	651
273	xfrac	651
193	lwarp-xltxtra.sty	653
274	xltxtra	653
194	lwarp-xmpinl.sty	654
275	xmpinl	654
195	lwarp-xtab.sty	654
276	xtab	654
196	lwarp-zwpagelayout.sty	656
277	zwpagelayout	656
	Change History and Index	658

List of Figures

1	tutorial.tex listing	58
---	----------------------	----

List of Tables

1	\LaTeX -HTML generation — lwarp package — Supported functions	43
2	Required software programs	51
3	Files created along with the print version	60
4	Package options	72
5	HTML settings	74
6	Section HTML headings for word-processor conversion	116
7	Section depths and HTML headings	125
8	Tabular baseline	306
9	Tabular HTML column conversions	306
10	Cross-referencing data structures	340
11	Float data structures	349
12	AMStm package — CSS styling of theorems and proofs	441
13	Ntheorem package — CSS styling of theorems and proofs	549
14	Theorem package — CSS styling of theorems and proofs	603

2 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 [658](#).

v0.44:

- [koma-script](#)
 - Added koma-script classes (except scrlltr2, scrjura).
 - Added scrextend, sclayer, sclayer-notecolumn, sclayer-scrpage, scrhack, tocstyle, tocbasic.
- [HTML title and author](#)
 - Added `\HTMLtitle`. Fixed web page title if `\HTMLtitle` empty and no `\title` given and not using titling package.
 - Fixed web page author if `\HTMLauthor` is empty and `\author` is not given.
- [encodings](#)
 - If using `pdflatex`, automatically loads T1 and UTF8 encodings. (Additional fontenc encodings may be loaded after lwarp.)
- [lists](#)
 - Added `list` and `trivlist` environments, `hang`.
- [tabular](#)
 - Fix: `\multicolumn` alignment if formatting for a word processor.
 - Added `ltxtable`.
- [math](#)
 - Fix: MathJax combined with `lateximages`.
 - `algorithmicx`: Improved comment symbol and floating.
- [packages](#)
 - Completed `todonotes` and `luatodonotes`.
 - Added `todo`, `easy-todo`, `fixmetodonotes`, `fixme`.
 - Added `soulutf8`, `soulpos`, `cancel`.
 - Added `section`, `fancyref`, `ifoddpage`.
 - Added `preview`, `atbegshi`, `watermark`.
 - Improved `tocloft` `\newlistof` and `\newlistentry`.

v0.43:

- Docs: Reorganized HTML customization, added an HTML settings table. See section [7.3](#).
- [footnotes](#)
 - Added `FootnoteDepth` to control the placement of pending footnotes before section breaks. By default, pending footnotes are printed before each `\subparagraph` or higher.
- [sectioning](#)
 - Fix: Expansion in section name.
- [tabular](#)
 - Fix: Ignore spaces in tabular column specification.
 - Fix: Tabular rules at bottom or when finishing incomplete rows.

math

- Fix: `\multicolumn` at/bang/before/after specifications, trim, and vertical rules.
- Fix: supertabular and xtab column misalignment.
- Fix: `equation*`.
- Fix: svg math in a section name.
- Fix: `\ref` and `\eqref` in svg math.

packages

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

v0.42:

Support T_EX!

- Added T_EX development support page.
- Improved assistance for word-processor conversions when boolean `FormatWP` is set true. See section 10.
 - The boolean `FormatWordProcessor` has been renamed `FormatWP`.
 - The boolean `HTMLMarkFloats` has been renamed `WPMarkFloats`.
 - New booleans control whether to place additional marks around minipages, at the table of contents, at the LOF and LOT, and whether to print math as \TeX source for copy/paste into the LibreOffice Writer TeXMaths extension.
 - Improved formatting for numerous objects. See section 10.

word-processor conversion

- ⚠ name change
- ⚠ name change

tabbing

- Add: tabbing environment.

overpic

- Add: overpic package. See section 202.

math

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

symbols

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

load order

- morewrites: Enforces loading before `lwarp`.

v0.41:

tabular

- Added tabular vertical rules, subject to some limitations. See the rules section of section 8.7.
- Improved booktabs: Width and trim are honored.

- ⚠ new syntax

- Added `\mcolrowcell` for empty cells inside a `\multicolumnrow`. Use `\mcolrowcell` instead of `\mrowcell` for two-dimensional cells created by `\multicolumnrow`. Continue to use `\mrowcell` for empty cells in a `\multirow`. See section 192.2 on page 545.

- 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

margins

columns

footnotes

tabular

sectioning

- 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 [8.6](#).
- grffile is now directly supported instead of emulated.
- Fix for bigdelim, and improved documentation. See section [101](#).
- Improved \TeX and textcomp symbols.
- Fix for \TeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for xltextra with $X_{\text{p}}\TeX$.
- 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`, `tabs`.
- Added `sectsty`, `anonchap`, `quotchap`.

v0.39:

title pages

⚠ `\published and`
`\subtitle`

⚠ load order

tabular

multi column/row cell

⚠ macros inside tabular

- 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 [55.7](#). `authblk` is added, and should be loaded before titling. See section [55](#).
- `\multirow` now supports the new optional `vpos` argument.
- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section [192.2](#).
- Tabular special cases:
 - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section [8.7](#).

⚠ tabular defined inside another environment

tabular
margins
page layout

– Added `\ResumeTabular` for use when a tabular environment is defined inside another environment. See section 8.7.

- Added `supertabular`, `xtab`, `bigstrut`, `bigdelim`.
- Added `fullwidth`.
- Added `addlines`, `anysize`, `a4`, `a4wide`, `a5comb`, `textarea`, `zwpagelayout`, `typearea`, `ebook`.

v0.38:

forced single-pass compile

starred sections

updated tutorial

packages
font size

page numbering

front & back matter

- 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 8.4.1.
- Modified the tutorial to use the new `\ForceHTMLPage` and `\ForceHTMLTOC` macros.
- Added `appendix`, `tocbibind`, `fncychap`, `fix2col`.
- Added `resize`, `scalegnt`.
- Added `realscripts`, `metalogo`, `xltxtra`.
- Added `grffile`, `romanbar`.
- Added `arabicfront`, `chappg`, `nonumonpart`, `nopageno`, `romanbarpagenumber`.
- Docs: Improved description of the use of front/back matter. See section 8.4.
- 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

- Recorganized the documentation section regarding special cases and limitations. (Section 8)
- 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 \TeX standards, and avoids problems with space between arguments.
- ⚠ **Optional arguments**
 - Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:


```
\InlineClass[style]{class}{text}
```
- spans with minipages**
 - Improved compatibility between `spans`, `minipages`, `lists`, `frames`, and `math`. Handles `minipages` and `lists` inside an `HTML` span, such as an `\fbox` containing a `minipage`, although with minimal `HTML` formatting. See section 8.2.3. `\fboxBlock` is added to frame `minipages`, `tables`, and `lists` with full `HTML` formatting but no longer inline, and behaves as `\fbox` for print output. The `fminipage` environment is added for framed `minipages`, as an environment with full `HTML` formatting, and draws a framed `minipage` in print output. See section 8.2.5. `\fbox` and `minipages` now often work in `SVG` `math` and `lateximages`. MathJax supports `\fbox`, but not `\fboxBlock` nor `fminipage`.
- lateximage, SVG math, tabular**
 - Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
- eqnarray**
 - Improved font control for `lateximagees` 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.
- Added rotfloat, marginfit, and several minor packages; see the change log.
- Adds fixed-width HTML spaces around punctuation when using babel-french. LuaTeX does not yet use the extra punctuation spacing.

babel-french

v0.33:

- Tabular @ and ! columns now have their own HTML columns.
- & catcode changes are localized, perhaps causing errors about the tab alignment character &, so any definitions of macros or environments which themselves contain tabular and & must be enclosed within \StartDefiningTabulars and \EndDefiningTabulars. See section 61.4.1. 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 ams-math 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 7.12.
- 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. Name changed to `FormatWP` as of v0.42.
- 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. Name changed to `WPMarkFloats` as of v0.42.
- Updated for the new MathJax CDN repository.
- Adds tabulary.
- Supports the options syntax for graphics.
- Improved index references, now pointing exactly to their target.
- Adds glossaries. `lwarpmk` is modified to add `printglossary` and `htmlglossary` actions.

3 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 HTML5, CSS3, and SVG are employed to allow a fairly feature-rich document without relying on the use of Javascript. Limited testing on older browsers show that these new features degrade gracefully, although the SVG format for math may not be available on small cell phones.

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

SVG images may be used for math, and are also used for `picture`, and `Tikz` environments, as this format has better browser and e-book support than MathML (as of this writing), while still allowing for the high-quality display and printing of images (again, subject to potentially bug-ridden¹ browser support).

Furthermore, SVG images allow math to be presented with the same precise formatting as in the print version. Math is accompanied by `<alt>` tags holding the \LaTeX source for the expression, allowing it to be copy/pasted into other documents.² Custom \LaTeX macros may be used as-is in math expressions, since the math is evaluated entirely inside \LaTeX .

The MATHJAX JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MathJax allows an HTML page to display math without relying on a large number of external image files, one per math expression. lwarp maintains \LaTeX control for cross-referencing and equation numbering / formatting.

The lwarp package allows \LaTeX to directly generate HTML5 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 HTML5 tags, to a number of HTML5 plain-text files and accompanying images.

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

²There seems to be some debate as to whether MathML is actually an improvement over \LaTeX for sharing math. The author has no particular opinion on the matter, except to say that in this case \LaTeX is much easier to implement!

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

When requesting packages during HTML conversion, lwarp first looks to see if it has its own modified version to use instead of the usual \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.)

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.

3.1 Supported packages and features

Supported classes include `book`, `report`, and `article`, and the Koma-script classes `scrbook`, `scrreprt`, and `scrtartcl`. `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> , <code>article</code> , <code>scrbook</code> , <code>scrreprt</code> , <code>scrtartcl</code> . <code>memoir</code> is planned.
Koma-script:	<code>scxextend</code> , <code>scrlayer</code> , <code>scrhack</code> . Others as listed below.

lwarp Supported Functions — continued

Category	Status
Page layout:	Emulates geometry, fancyhdr, titleps, sclayer-scrpage, typearea, addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, ebook, preview, draftwatermark, watermark, everyshi, atbegshi.
Sectioning:	Supported, with hyperlinks. Adds FileDepth 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 titlesec, fncychap, sectsty, section, anonchop, quotchap.
Table of contents, figures, tables:	Supported, with hyperlinks. Supports tocibind. Emulates titletoc, tocloft, tocbasic, and tocstyle.
Title page:	\maketitle, titlepage, titling, authblk.
Front & back matter:	abstract, appendix
Indexing:	texindy is used, with hyperlinks. idxlayout is emulated.
Glossary:	glossaries and xindy are used.
Bibliography:	Supported text-only.
Cross-references:	Emulated, with hyperlinks. hyperref, cleveref, varioref, fancyref, url, breakurl.
Languages:	babel. (polyglossia is untested.)
Margin notes:	marginfit, marginfix, sclayer-notecolumn.
Footnotes:	Adds FootnoteDepth to print footnotes at section breaks. footnote, footmisc, marginnote, sidenote, pagenote, endnotes.
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 theorem, amsthm, ntheorem.

lwarp Supported Functions — continued

Category	Status
Add'l math:	delarray, bm, math fonts via SVG images.
Units and fractions:	siunitx, xfrac, nicefrac, units
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, ltxtable, 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. rotating is emulated but all objects are unrotated. picture and tikz are converted to an SVG image. grffile and overpic are supported.
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 L ^A T _E X environments are supported, along with enumitem, enumerate, paralist, hang.
Environments:	Standard L ^A T _E X environments are supported.
minipage:	Supported with some HTML5-imposed limitations. Nested minipages are supported.
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 <code>css3</code> . Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
fullwidth:	Emulated.
Todo:	<code>todo</code> , <code>todonotes</code> , <code>easy-todo</code> , <code>fixmetodonotes</code> , <code>fixme</code> .
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>soulutf8</code> , <code>soulpos</code> , <code>cancel</code> , <code>reysize</code> , <code>scaleftnt</code> , and <code>realscripts</code> are supported.
Ordinals:	<code>nth</code> , <code>fntcount</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 <code>width</code> , <code>height</code> , <code>raise</code> , <code>text color</code> .
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, <code>file</code> , 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.

4 Alternatives

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

4.1 Internet class

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

4.2 TeX4ht

Prog `TeX4ht` <http://tug.org/tex4ht/>
 Prog `htlatrix` 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 most cases, lwarp provides a better HTML conversion, while supporting more packages. TeX4ht produces several other forms of output beyond HTML.

4.3 Translators

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

Prog `Hevea` **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` **\LaTeX XML**: <http://dlmf.nist.gov/LaTeXML/>
 Prog `Plastex` **PlasTeX**: <https://github.com/tiarno/plastex>
 Prog `LaTeX2HTML` **\LaTeX 2HTML**: <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 \LaTeX math into HTML:

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

4.4 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 \LaTeX 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 \LaTeX -related features.

Asciidoctor-LaTeX:

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

4.5 Pandoc

Prog Pandoc

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

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

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

4.6 Word processors

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

4.7 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

Prog Flare

Prog Madcap

4.8 Comparisons

AsciiDoc, Pandoc, and various other markup languages typically have a syntax which tries to be natural and human-readable, but the use of advanced features tends to require many combinations of special characters, resulting in a complicated mess of syntax. By contrast, \LaTeX spells things out in readable words but takes longer to type, although integrated editors exist which can provide faster entry and a graphic user interface. For those functions which are covered by the typical markup language it is arguable that \LaTeX is comparably easy to learn, while \TeX 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 \TeX back end, yielding high-quality results especially when the \TeX template is adjusted, but they lose the ability to use \TeX macros and other \TeX source-document features.

The effort required to customize the output of each markup system varies. For print output, \TeX 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 \TeX optional macro parameters, but further customization may require patching \TeX code.

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

5 Installation

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

5.1 Installing the lwarp package

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

Pre-installed: Try entering into a command line:

```
Enter ⇒ kpsewhich lwarp.sty
```

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

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

```
Enter ⇒ tlmgr install lwarp
```

Mi \TeX : If using Mi \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
```

Mi \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 TeXLive: <http://tug.org/texlive/>.

\LaTeX : pdflatex, xelatex, or lualatex.

The lwarp package: This package.

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

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

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

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

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

Poppler PDF utilities:

pdftotext: Used to convert PDF to text.

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

pdftocairo: Used to convert images to SVG.

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

From Poppler: poppler.freedesktop.org.

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

```
Enter ⇒ brew install poppler
```

For Windows, see:

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

Perl:

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

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 ⇒ mktexlsr
```

 — 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 5.2.1 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 5.2.

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

Just testing!

5.2 Installing the lwarpmk utility

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

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

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

Otherwise, continue with the following:

3. Locate the file lwarpmk.lua, which should be in the scripts directory of the TDS tree. On a 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 5.2.1.)

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.

5.2.1 Using a local copy of `lwarpmk`

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

1. When compiling the tutorial in section 6, use the `lwarpmk` option for the `lwarp` package:

```
\usepackage[lwarpmk]{lwarp}
```

2. When the tutorial is compiled with `pdflatex`, the file `lwarpmk.lua` will be generated along with the other configuration files.
3. `lwarpmk.lua` may be used for this project:

Unix:

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

Windows:

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

```
Enter ⇒ texlua lwarpmk.lua html
```

```
Enter ⇒ texlua lwarpmk.lua print
```

etc.

Or:

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print
```

etc.

5.3 Installing additional utilities

To test for the existence of the additional utilities:

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 2 on page 51.

```
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 51.
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\
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.

6 Tutorial

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

6.1 Starting a new project

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

File `tutorial.tex`

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

Copy from the documentation PDF:

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

Copy from the lwarp documentation directory:

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
\HTMLTitle{Webpage Title} % Overrides \title for the web page.
\HTMLAuthor{Some Author} % Sets the HTML meta author tag.
\HTMLLanguage{en-US} % Sets the HTML meta language.
\HTMLDescription{A description.} % Sets the HTML meta description.
```

```

\HTMLFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}
\HTMLPageTop{\fbox{LOGO}}
\HTMLPageBottom{Contact Information and Copyright}
\CSSFilename{lwrap_sagebrush.css}

\begin{document}

\maketitle % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

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

\end{document}

```

Table 3: Files created along with the print version

- tutorial.pdf:** The PDF output from \LaTeX . The print version of the document.
- tutorial_html.tex:** A small `.tex` file used to create a parallel HTML version of the document, which co-exists with usual the PDF version, and which will have its own auxiliary files. In this way, both PDF and HTML documents may co-exist side-by-side.
- Auxiliary files:** The usual \LaTeX files `.aux`, `.log`, `.out`, `.toc`, `.lof`, `.idx`. When an HTML version of the document is created, `_html` versions of the auxiliary files will also be generated.
- lwarpmk.conf:** A configuration file for `lwarpmk`, which is used to automate the compilation of PDF or HTML versions of the document.
- tutorial.lwarpmkconf:** Another configuration file used by `lwarpmk`, which is only useful if you wish to have several projects residing in the same directory.
- .css files:** `lwarp.css`, `lwarp_formal.css`, `lwarp_sagebrush.css` These files are standard for `lwarp`, and are not meant to be modified by the user.
- sample_project.css:** An example of a user-customized css file, which may be used for project-specific changes to the `lwarp` defaults.
- lwarp.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.

6.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

```
Enter ⇒ lwarpmk print
```

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

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

```
Enter ⇒ lwarpmk print
```

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

4. Force a recompile to occur.

```
Enter ⇒ lwarpmk again
```

```
Enter ⇒ lwarpmk print
```

`lwarpmk again` updates the date code for the file, triggering a recompile the next time the document is made.³

5. Process the index.⁴⁵

```
Enter ⇒ lwarpmk printindex
```

6. Recompile again to include the index.

```
Enter ⇒ lwarpmk print
```

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

```
Enter ⇒ lwarpmk print1
```

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

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

⁴A `lwarpmk printglossary` command is also available to process a glossary produced with the glossaries package. See section 8.4.8.

⁵Also see section 8.4.9 for index options.

6.3 Compiling the HTML version with lwarpmk

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

1. Compile the HTML version:

```
Enter ⇒ lwarpmk html
```

- (a) lwarpmk uses \LaTeX to process tutorial_html.tex to create tutorial_html.pdf.
- (b) pdftotext is then used to convert to the file tutorial_html.html. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) lwarpmk manually splits tutorial_html.html into individual HTML files according to the HTML settings. For this tutorial, the result is tutorial.html (the home page), along with First-chapter.html⁶, 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 6.4 and 6.5.

3. Force a recompile:

```
Enter ⇒ lwarpmk again
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print
```

4. Process the HTML index and recompile: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
```

⁶First-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.

⁷index.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 8.4.8.

⁹Also see section 8.4.9 for index options.

6.4 Generating the SVG images

math as svg images By default lwarp represents math as svg images with the \LaTeX source included in `alt` attributes. In this way, the math 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.

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 `$$` is used, for example, yet another file is created. For a document with a large amount of math, see section 6.5 to use MathJax instead.

 **Lots of files!**

6.5 Using MathJax for math

[math with MathJax](#) Math may also be represented using the MathJax Javascript project.

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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.

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

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

6.6 Changing the CSS style

For a formal css style, add to the preamble:

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

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

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

See section [7.4](#) for more information about modifying the CSS styling of the document.

6.7 Customizing the HTML output

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

See section [7.3](#) for more information.

6.8 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
```

6.9 Using XeLaTeX or LuaLaTeX

X_εLaTeX or LuaLaTeX may be used instead of L^AT_EX.

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_εLaTeX or LuaLaTeX will be used for future runs of `lwarpmk`.

3. To recompile the document:

```
Enter ⇒ lwarpmk print
```

-and-

```
Enter ⇒ lwarpmk html
```

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

6.10 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.

6.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

6.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

6.13 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
```

6.14 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.

7 Additional details

7.1 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 older bit-mapped fonts.

`pdflatex`, T1, UTF8 While using `pdflatex`, `fontspec` is automatically loaded with T1 encoding, and `inputenc` is automatically loaded with UTF8 encoding, each of which is required for the conversion process. `fontspec` may be loaded with an additional encoding after `lwarp`.

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:

- `fontspec` and font choices

Pkg `fontspec`

`ligatures`

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

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

- For `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg	fontenc	- fontenc
Pkg	inputenc	- inputenc
Pkg	newunicodechar	- newunicodechar
File	glyphtounicode	- \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 \TeX ligatures.
	ligatures	

3. `\usepackage{lwarp}` (section 7.2) goes after any of the above, followed by:
4. ... the rest of the preamble and the main document.

7.1.1 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 \TeX . This option correctly sorts index entries into headings while using Latin languages, but will not work well with others. $X_{\text{e}}\TeX$ or $\text{Lua}\TeX$ are recommended for non-Latin languages.

For an older distribution of \TeX , it may be necessary to generate a local version of `lwarpmk.lua` and modify it to remove the `-C utf8` option from the `texindy` call. See section 11.4.

Table 4: Package options

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

7.2 lwarp package loading and options

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

Pkg	<code>lwarp</code>	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	<code>warpprint</code>	Usually controlled by <code>lwarpmk</code> , and not set in the document. Select the <code>warpprint</code> option to generate print output (default), or the <code>warpHTML</code> option to generate HTML5 output. The default is print output, so the print version may be compiled with the usual <code>pdflatex</code> , etc. When lwarp is loaded in print mode, it creates <code><project>_html.tex</code> , which sets the <code>warpHTML</code> option before calling the user's source code <code><project>.tex</code> . In this way, <code><project>.tex</code> can <code>\usepackage{lwarp}</code> without any options to create a printed version, while <code><project>_html.tex</code> will create an HTML version.
Opt	<code>warpHTML</code>	
Opt	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 8.5.
Opt	<code>mathjax</code>	

-
- Opt `OSWindows` See section 7.5 if using Windows.
- Opt `BaseJobname` Not intended for the user. Used internally by lwarp when creating the `*_html.tex` file used to compile the HTML version. See section 21.
- Opt `HomeHTMLFilename` See section 7.3.
- Opt `HTMLFilename` See section 7.3.
- Opt `IndexLanguage` If using an index or glossary, see section 21.
- Opt `latexmk` Has `lwarpmk` use `latexmk` to recompile the document several times if necessary. Otherwise, `lwarpmk` attempts to determine for itself whether to recompile. See section 7.3.
- Opt `lwarpmk` If you wish to have lwarp generate a local copy of `lwarpmk.lua` for archival or local-installation purposes, compile the print version with the `lwarpmk` option set. See section 21.
- Opt `xdyFilename` The default `xindy` filename is `lwarp.xdy`. If you wish to use a custom `.xdy` file for index generation, see section 21.

Table 5: HTML settings

Option	Description
SideTOCDepth	Sectioning depth of the sideroc.
FileDepth	Sectioning depth of the file splits.
CombineHigherDepths	Combine higher section levels.
FileSectionNames	Use section names for file names, else use numbers.
FootnoteDepth	Sectioning depth of footnotes.
\abstractname	The name of the abstract.
\CSSFilename	The css for the following files.
\HTMLLanguage	The html lang tag.
\HTMLTitle	The HTML title meta tag, overriding \title.
\HTMLAuthor	The HTML author meta tag, overriding \author.
\HTMLDescription	The HTML description meta tag.
\HTMLFirstPageTop	Heading for the home page.
\HTMLPageTop	Heading for the other pages.
\HTMLPageBottom	Footing for all pages.

7.3 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 ⇒ lwarpmk clearall
```

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

```
Enter ⇒ lwarpmk print
```

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

```
Enter ⇒ lwarpmk html
```

Options for the lwarp package:

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

Opt `HomeHTMLFilename` **HomeHTMLFilename:** Filename of the homepage, without the “.html” suffix.
 Default: `\BaseJobname` 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}] \`

See section 7.3.1 for examples of naming and numbering HTML files.

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. See section 7.3.1 for examples of naming and numbering HTML files.
 Default: `<empty>`

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

Opt `mathsvg` **mathsvg:** Selects svg display for math output. (The default.)

Default: `true`

Opt `mathjax`

mathjax: Selects MathJax for math output.

Default: `false`

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

Ctr `tocdepth` **tocdepth:** Sectioning depth of the table of contents. See section 14 for a list of L^AT_EX stack depths.

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

sideroc

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`

⚠ inaccessible pages

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

Ctrl `FileDepth`
 Default: -5

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

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



Bool `CombineHigherDepths`
 Default: true

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

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

FileSectionNames: If true, web page filenames are derived from a sanitized version of the section names. If false, web pages are numbered. Either way, the `HTMLFilename` option is used as a prefix. See section 7.3.1 for examples of naming and numbering HTML files.

Ctrl `FootnoteDepth`
 Default: 5

FootnoteDepth: Determines where to place pending footnotes. 5 places footnotes before each break down to the `\ subparagraph` level. 1 places footnotes before each `\section` break. Any pending footnotes are also placed at the bottom of each page before each file break.

`\abstractname`
 Default: Abstract

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

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

`\CSSFilename`
Default: `lwarp.css`

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

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

`\HTMLLanguage`
Default: `en-US`

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

`\HTMLTitle`
Default: `\thetitle`

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

`\HTMLAuthor`
Default: `\theauthor`

`\HTMLAuthor: {\author}` The HTML header's meta author. Defaults to `\theauthor`, which is set by `\author` if using the titling package, but is empty otherwise. There are several ways to represent the author and affiliations, especially if using the `authblk` package, most of which do not result in a sensible `\theauthor`, so `\HTMLAuthor` is useful to create a list of authors without their affiliations.

`\HTMLDescription`
Default: `<empty>`

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

`\HTMLFirstPageTop`
Default: `<empty>`

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

`\HTMLPageTop`
Default: `<empty>`

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

`\HTMLPageBottom`
Default: `<empty>`

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

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

 `\tableofcontents`
TOC on the homepage!

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

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

Placed in the document wherever necessary:

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

7.3.1 Example HTML file naming

Examples of ways to name or number HTML files:

Numbered HTML nodes:

Example: Homepage `index.html`, and `node-1`, `node-2`.¹⁰

```

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

```

¹⁰See `\SetHTMLFileNumber` to number in groups by chapter, for example.

Named HTML sections, no prefix:

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

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

Named HTML sections, with prefix:

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

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

7.4 Customizing the CSS

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

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

The styles provided by lwarp include:

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

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

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

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

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

 **Rename it!**

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

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

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

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 --- ) */

```

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

7.5 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 `OSWindows`.

Prog `Windows`

Opt `OSWindows`

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.

7.6 Selecting actions for print or HTML output

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

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

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

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

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

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.

7.7 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 [12: Troubleshooting](#).

7.8 Title page

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

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

The title is used in the meta tags in the HTML files, unless overridden by `\HTMLTitle`, and the rest are used in `\maketitle`. To use a `\subtitle` or `\published` field, see section [55.7](#).

`\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 55.7). The title will appear in the document `\maketitle` as a heading `<h1>`. The HTML meta `title` tag will also have this title, unless `\HTMLTitle` is used to set the meta title to something else instead.

`\author` `{<author>}`



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

`\date` `{<date>}`

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

`\thanks` `{<text>}`

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

7.9 HTML page meta descriptions

`\HTMLDescription` `{<A description of the web page.>}` The default is no description.

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

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

7.10 HTML page meta title

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

7.11 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.

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

7.12 Modifying xindy index processing

Prog `xindy` `lwarp` uses the file `lwarp.xdy` to process the index. This file is over-written by
File `lwarp.xdy` `lwarp` whenever a print version of the document is processed.

To customize index processing:

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


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

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

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

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

4. Recompile the print version, which causes `lwarp` to rewrite the `lwarpmk.conf` configuration file. This tells `lwarpmk` to use the custom `projectname.xdy` file instead of `lwarp.xdy`.

8 Special cases and limitations

Also see section 12: [Troubleshooting](#).

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

8.1 Formatting

8.1.1 Text formatting

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

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

8.1.2 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 `\qqquad`.

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

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

8.1.3 Text alignment

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

8.1.4 Accents

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

8.1.5 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.

8.1.6 Superscripts and other non-math uses of math mode

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

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

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

8.1.8 Filenames and URLs in lists or footnotes

filename underscore Escape underscores in the filenames:

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

8.1.9 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.

8.2 Boxes and minipages

8.2.1 Marginpars

`\marginpar` [*left*] {*right*} `\marginpar` may contain paragraphs, but in order to re-

main inline with the surrounding text lwarp nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `
` tags.

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

8.2.2 Save Boxes

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

8.2.3 Minipages

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

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

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

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

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

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

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

 **text alignment** Nested minipages adopt their parent’s text alignment in HTML, whereas in regular \TeX PDF output they do not. Use a `flushleft` or similar environment in the child

minipage to force a text alignment.

8.2.4 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.

8.2.5 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:

`\fbox`
`\fboxBlock`
 Env `fminipage`

For a framed object, options include:

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

For inline text:

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

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

For inline minipage and lists:

To create a framed minipage in both print and HTML: A new environment `fminipage` is included. For print output, this is identical to `minipage`, except

For display `tabular`,
 minipages, and lists:

that it is also framed. For HTML output, this forms a `<div>` of class `framed`, the contents preserve their HTML formatting, and valid HTML is created for a tabular. Also see below regarding the “Misplaced alignment tab character &” error.

colored boxes and frames: **To create colored frames and boxes:** See section 272 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 133 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}
```

8.2.6 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.

8.2.7 `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.

8.3 Cross-references

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

⚠ **underscores** does not yet work when generating math for MathJax.

`\nameref` `\nameref` refers to the most recently-used section where the `\label` was defined. If

⚠ **empty link** no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.

8.3.1 Page references

⚠ **TeX 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 345.

8.3.2 **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 402 to redefine the message which is printed for page number references.

8.3.3 **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.

8.3.4 **Footnotes and page notes**

lwarp uses native \TeX footnote code, although with its own `\box` to avoid the \TeX 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.

8.4 **Front and back matter**

8.4.1 **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.

8.4.2 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.

8.4.3 titling and authblk

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 55.7.

8.4.4 **tocloft** package

 **tocloft & other packages** `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 \TeX commands to create the titles, allowing other packages to work with it.

8.4.5 **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.

8.4.6 **pagenote** package

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

8.4.7 **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
```

8.4.8 **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`

8.4.9 Index and the `tocbibind` 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 `tocbibind`, `anonchp`, `fncychap`, or other packages which change chapter title formatting, load `tocloft` with its `titles` option, which tells `tocloft` to use standard \TeX commands to create the titles, allowing other packages to work with it.

[placement and roc 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 \TeX 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 [65](#) for `lwarp`'s core index and glossary code, and section [258](#) for `tocbibind`.

8.5 Math

8.5.1 Rendering tradeoffs

[Math rendering](#) Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.

[SVG files](#) In its current implementation, rendering math as images creates a new SVG file for each expression. 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 check-summing may be used to remove the need for duplicate files.

[SVG inline](#) Another approach could be to in-line the SVG files directly into the HTML. This may

reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.

PNG files Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but svg files are the preferred approach for scalable graphics.

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than svg drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

8.5.2 SVG option

SVG math option For svg math, math is rendered as usual by \TeX 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 \TeX 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 \TeX code which generated the math, allowing copy/paste of the \TeX 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 math copy/paste For svg math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the \TeX code for the math expression. For \mathcal{AMS} environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred \mathcal{AMS} environment, although the `\tag` macro will still appear inside the \TeX math expression.

8.5.3 MathJax option

MathJax math option The popular MathJax alternative (mathjax.org) may be used to display math.

Prog MathJax

When MathJax is enabled, math is rendered twice:

1. As regular \TeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of \TeX , and

¹¹See section 273 regarding fonts and fractions.

- 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

Prog MathJax

Limitations when using MathJax include:

chapter numbers

- In document classes which have chapters, `\tagged` equations have the chapter number prepended in HTML output, unlike \LaTeX . `\tag*` equations (correctly) do not. This may be improved with future versions of the MathJax support script.

<https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY>

subequations

- MathJax itself does not support subequations. This may be improved by parsing the \LaTeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

- Footnotes inside equations are not yet supported while using MathJax.

lateximage

- Math appearing inside a `lateximage`, and therefore also inside a `Tikz` or `picture` environment, is rendered as SVG math even if MathJax is used in the rest of the document.

siunitx

- Usage of `siunitx` inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

<https://github.com/burnpanck/MathJax-siunitx>

Also see section 8.5.5.

siunitx inside an equation

other macros and packages

- Other math-related macros and packages are not supported by MathJax, including `\ensuremath`, `bigdelim`, `units`, and `nicefrac`, along with occasionally-used macros such as `\footnote` and `\relax`.

custom MathJax macros

- MathJax does not automatically support custom \LaTeX macros, but they may be created by the user inside a math expression:

```

\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}

```

8.5.4 ntheorem package

 **Font control** This conversion is not total. Font control is via css, and the custom \LaTeX font settings are ignored.

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

8.5.5 siunitx package

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

 **per-mode**

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

NOTE: As of this writing, the `siunitx` extension for MathJax is not currently hosted at any public CDN, thus `siunitx` is not usable with MathJax unless a local copy of this extension is created first.

8.5.6 units and nicefrac packages

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

8.5.7 newtxmath package

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

```

...
\usepackage{lwarp}
...
\usepackage{amsthm}
\usepackage{newtxmath}
...

```

8.6 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 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 com-

bination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

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

⚠ **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \TeX , so expect some ugly results for scaling and rotating.

8.6.1 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.

8.6.2 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.

8.6.3 xcolor package

`\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.

8.6.4 overpic package

- △ **scaling** The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

8.7 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 192.2 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.

⚠ `\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:

- `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 \\\}
```

8.7.1 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`.

8.7.2 supertabular and xtab packages

⚠ **misplaced alignment** For `\tablefirsthead`, etc., enclose them as follows:
alignment tab character &

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

See section 8.7.

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

```

	a	b
left	{	c d
		e f
		g h

8.8 Floats

8.8.1 float, trivfloat, and/or algorithmicx together

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [264.1](#).

8.8.2 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`.

8.8.3 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.

8.8.4 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.

The emulation of `floatrow` does not support `\FBwidth` or `\FBheight`. These values are pre-set to `.3\linewidth` and `2in`. Possible solutions include:

⚠ `\FBwidth`, `\FBheight`

- 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.

8.8.5 keyfloat package

⚠ `keywrap` If placing a `\keyfig[H]` inside a `keywrap`, use an absolute width for `\keyfig`, instead of `lw`-proportional widths. (The `[H]` option forces the use of a `minipage`, which internally adjusts for a virtual 6-inch wide `minipage`, which then corrupts the `lw` option.)

8.9 Koma-Script

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

`\titlehead`, `\subject`, `\captionformat`, `\figureformat`, and `\tableformat` are not yet emulated.

⚠ Not yet tested! [Please send bug reports!](#)

Some features have not yet been tested. Please contact the author with any bug reports.

8.10 Miscellaneous

8.10.1 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.

8.10.2 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>
--	---

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

8.10.3 babel package

Pkg `babel`

`\CaptionSeparator` When French is used, the caption separator is changed to a dash. The following may be used to restore it to a colon:

```
\renewcommand*\CaptionSeparator}{:~}
```

punctuation spaces Also when French is used, lwarp creates fixed-width space around punctuation by patching `\FBcolonspace`, `\FBthinspace`, `\FBguillspace`, `\FBmedkern`, `\FBthickkern`, `\FBtextellipsis`, and the tilde. If the user's document also changes these parameters, the user's changes should be placed inside a `warpprint` environment so that the user's changes do not affect the HTML output.

⚠ **customized spacing**

8.10.4 **todonotes and luatodonotes packages**

The documentation for `todonotes` and `luatodonotes` have an example with a `todo` inside a caption. If this example does not work it will be necessary to move the `todo` outside of the caption.

8.10.5 **fixme**

⚠ **external layouts** External layouts (`\fxloadlayouts`) are not supported.

User control is provided for setting the HTML styling of the “faces”. The defaults are as follows, and may be changed in the preamble after `fixme` is loaded:

```
\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}
```

9 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:

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

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

```
Enter ⇒ lwarpmk cleanall
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk limages
```

to recompile with the `FormatEPUB` boolean turned on. Several changes are then made to the HTML output:

- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.

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 7. For example, an `article` class document would break at `\section`, which is mapped to HTML heading level `<h4>`, whereas a `book` class document would break at `\chapter`, which is HTML heading level `<h3>`. For Calibre, this option is found in

Preferences → Conversion (Common Options) → Structure Detection → Detect chapters at (XPath expression)

Select the “magic wand” to the right of this entry box, and set the first entry

Match HTML tags with tag name:

to “h4”. (Or “h3” for document classes with `\chapters`.) The Detect chapters at field should then show

`//h:h4` — or — `//h:h3`

This option is also available on the main tool bar at the Convert books button.

Once these settings have been made, the lwarp-generated HTML files may be loaded by Calibre, and then converted to an EPUB.

MathJax support

MathJax may be used in EPUB documents. Some e-readers include MathJax, but any given reader may or may not have a recent version, and may or may not include extensions such as support for `siunitx`.

lwarp adds some modifications to MathML to support equations numbered by chapter. These modifications may not be compatible with the e-reader's version of MathJax, so lwarp requests that a known version be loaded instead. In some cases chapter numbering of equations still doesn't work.

Until math support in EPUB documents is improved, it is recommended to use SVG images instead of MathJax, especially for equations numbered by chapter, or where `siunitx` support is important.

10 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.

10.1 Activating word-processor conversion

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

FormatWP

Bool `FormatWP`
Default: `false`

Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments. Additionally, honors the booleans `WPMarkFloats`, `WPMarkMinipages`, `WPMarkTOC`, and `WPMarkLOFT`.

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

```
\booltrue{FormatWP}
```

formatting adjustments

to the project's source preamble after lwarp is loaded. The following changes are then made to the HTML output:

- If using a class without chapters, `\section` and lower are shifted up in level for the HTML heading tags. The CSS has not been changed, so the section heading formats will not match the normal HTML output, but when imported to LibreOffice Writer the higher section headings will import as **Heading 1** for the title, **Heading 2** for `\section`, etc.
- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each section.
- Forces single-file output.
- Turns off HTML debugging comments. These are comments appearing inside the HTML code, marking the opening/closing of sections and `<div>`s, but they are no longer useful when the document has been imported into a word processor.

- An additional `<div>` with an `id` encapsulates each float and minipage, which on import into LibreOffice Writer causes a thin frame to appear around the text block for each.
- Float captions are given an explicit italic formatting.
- Tabular rule borders are made explicit for LibreOffice Writer. LibreOffice displays a light border around each cell while editing, even those which have no border when printed, and lwarp also uses a light border for thin rules, so it will be best to judge the results using the print preview instead of while editing in LibreOffice.
- `\includegraphics` and `svg` math width and height are made explicit for LibreOffice.
- `\hspace` is approximated by a number of `\quads`, and rules are approximated by a number of underscores.
- Explicit HTML styles are given to:
 - `\textsc`, etc.
 - `\underline`, `soul` and `ulem` markup.
 - `center`, `flushleft`, `flushright`.
 - `\marginpar`, `keyfloat`, `sidenotes`, `floatflt`, and `wrapfig`.
 - `fancybox` `\shadowbox`, etc.
 - The \LaTeX and \TeX logos.
- Honors several booleans:
 - WPMarkFloats:** Marks the begin and end of floats.
 - WPMarkMinipages:** Marks the begin and end of minipages.
 - WPMarkTOC:** Marks the location of the Table of Contents.
 - WPMarkLOFT:** Marks the locations of the List of Figures/Tables.
 - WPMarkMath:** Prints \LaTeX math instead of using images.
 - WPTitleHeading:** Adjusts title and section headings.

Several of these may be used to add markers to the HTML text which help determine where to adjust the word processor document after import.

10.2 Additional modifications

WPMarkFloats

Adds

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

or

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

Bool `WPMarkFloats`
 Default: `false`

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

WPMarkMinipages

Adds

```
=== begin minipage ===
...
=== end minipage ===
```

Bool `WPMarkMinipages`
 Default: `false`

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

WPMarkTOC

While formatting for word processors, adds

```
=== table of contents ===
```

Bool `WPMarkTOC`
 Default: `true`

where the Table of Contents would have been. This helps identify where to insert the actual TOC.

If set false, the actual TOC is printed instead.

WPMarkLOFT

While formatting for word processors, adds

```
=== list of figures === and/or
=== list of tables ===
```

Bool WPMarkLOFT
Default: false

where each of these lists would have been. This helps identify where to insert the actual lists.

If set false, the actual lists are printed instead.

WPMarkMath

While formatting for word processors, prints math as \LaTeX code instead of creating svg images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

Bool WPMarkMath
Default: false
Prog TeXMaths
siunitx

When using the siunitx package, enter

```
\usepackage{siunitx}
```

in the TeXMaths preamble. Equation numbering is problematic for \mathcal{AMS} math environments.

WPTitleHeading

While formatting for word processors, true sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to false to cause the title to be plain text, and the section headings to begin at **Heading 1**.

Bool WPTitleHeading
Default: false
section headings

See table 6 on page 116.

10.3 Recommendations

TOC, LOE, LOT For use with LibreOffice Writer, it is recommended to:

1. Set `\booltrue{FormatWP}`.
2. Set `\booltrue{WPMarkTOC}` and `\boolfalse{WPMarkLOFT}`.
3. Use lwarp to generate the HTML document.
4. Copy/paste from the HTML document into an empty LibreOffice Writer document.
5. Manually insert a LibreOffice TOC in the LibreOffice document.

Table 6: Section HTML headings for word-processor conversion

Section	HTML headings*			
	With <code>\chapter</code>		Without <code>\chapter</code>	
	WPTitleHeading		WPTitleHeading	
	true	false	true	false
Title	<code><h1></code>	plain	<code><h1></code>	plain
<code>\part</code>	<code><h2></code>	<code><h1></code>	<code><h2></code>	<code><h1></code>
<code>\chapter</code>	<code><h3></code>	<code><h2></code>	—	—
<code>\section</code>	<code><h4></code>	<code><h3></code>	<code><h3></code>	<code><h2></code>
<code>\subsection</code>	<code><h5></code>	<code><h4></code>	<code><h4></code>	<code><h3></code>
<code>\paragraph</code>	<code><h6></code>	<code><h5></code>	<code><h5></code>	<code><h4></code>
<code>\subparagraph</code>	span	<code><h6></code>	<code><h6></code>	<code><h5></code>

* For default depths when not FormatWP, see table 7 on page 125.

6. Manually add frames around each float, adding a caption which is cut/pasted from each float's simulated caption.
7. Manually create cross references.

This process yields a document with an actual LibreOffice Table of Contents, but a simulated List of Figures and List of Tables.

[siunitx](#) For siunitx, remember to adjust the preamble as mentioned above.

[LO view border options](#) LibreOffice has options in the View menu to turn on/off the display of thin borders around table cells and text objects.

10.4 Limitations

Floats and captions are not explicitly converted to LibreOffice floats with their own captions. Floats are surrounded by a thin frame in the LibreOffice editor, and may be marked with `WPMarkFloats`, but are not given a proper LibreOffice object frame. Captions are given an explicit italic formatting, but not a proper LibreOffice paragraph style.

Cross references are not actual LibreOffice linked cross references.

The List of Figures and List of Tables are not linked. The pasted pseudo LOF and LOT match the numbering of the \LaTeX and HTML versions.

Equation numbering is not automatic, but the equation numbers in SVG math will match the \LaTeX and HTML output. SVG math is recommended when using the \AMS environments, which may have multiple numbered equations per object.

As of when last checked, LibreOffice ignores the following:

- Minipage alignment.
- Tabular cell vertical alignment.
- Image rotation and scaling.
- Rounded border corners, which are also used by:
 - `\textcircled`
 - `booktabs trim`
- `\hspace` and `rules`, also used by `algorithmic`.
- Coloring of text decorations, used by `soul` and `ulem`.
- Overline text decoration, used by `romanbar`.

Libreoffice also has limitations with frames and backgrounds:

- Multiple lines in an object are framed individually instead of as a whole.
- Nested frames are not handled correctly.
- Images inside boxes are not framed correctly.
- Spans with background colors and frames are not displayed correctly.

11 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 75 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 42.8.

11.1 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 201 (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.

11.2 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 22. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`.

11.3 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.

11.4 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`”.

12 Troubleshooting

12.1 Using the `lwarp.sty` package

Also see:

Section 7.7: [Commands to be placed into the warpprint environment](#)

Section 8: [Special cases and limitations](#)

Text is not converting:

- Font-related UTF-8 information must be embedded in the PDF file. See section 7.1 regarding vector fonts.

Undefined HTML settings:

- See the warning regarding the placement of the HTML settings at section 7.3.

Tabular problems: See section 8.7.

Obscure error messages:

Print first: Be sure that a print version of the document compiles and that your document's \LaTeX code is correct, before attempting to generate an HTML version.

“Missing \$ inserted.”: If using a filename or URL in a footnote or `\item`, escape underscores with `_.`

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

“Improper `\prevdepth`”: Something tried to use `\ensuremath` where `lwarp` then tries to create a `lateximage`. If you can locate what used `\ensuremath`, have it temporarily set:

```
\LetLtxMacro\@ensuredmath\LWR@origensuredmath
inside a group first.
```

Missing sections:

- See section 7.3 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 7.3.

Missing / incorrect cross-references:

labels

⚠ underscores

`\nameref`

⚠ empty link

⚠ `cleveref` page numbers

- Use `lwarpmk` again followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.
- Labels with special characters may be a problem. It is best to stick with alpha-numeric, hyphen, and perhaps the colon (if not French). The underscore currently does not yet work when generating math for MathJax.

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

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

Ex:

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

in HTML becomes:

“pages **for** table 4.1 and **for** table 4.2”

See `\cpagerefFor` at page 402 to redefine the message which is printed for page number references.

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

Use \TeX or \LaTeX .

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 `[warpHTML]`.

Images are appearing in strange places:

- `lwarpmk limages` to refresh the `lateximage` images.

SVG images:

⚠ 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 6.5 to use MathJax instead.

Plain-looking document:

- The document's CSS stylesheet may not be available, or may be linked incorrectly. Verify any `\CSSFilename` statements point to a valid CSS file.

Broken fragments of HTML:

- Check the PDF file used to create HTML to see if the tags overflowed the margin. (This is why such large page size and margins are used.)

Changes do not seem to be taking effect:

- Be sure to `lwarpmk clean`, recompile, then start by reloading the home page. You may have been looking at an older version of the document. If you changed a section name, you may have been looking at the file for the old name.
- See the warning regarding changes to the HTML settings at section 7.3.
- 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.

12.1.1 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.

12.2 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}`

File 1 **lwarp.sty**

13 Implementation

This package is perhaps best described as a large collection of smaller individual technical challenges, in many cases solved through a number of ~~erude hacks~~ clever tricks. Reference sources are given for many of the solutions, and a quick internet search will provide additional possibilities.

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

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

Packages were emulated where their primary purpose was visual formatting which is not relevant to HTML output. For example, packages related to sectioning are already patched by numerous other packages, creating a difficult number of combinations to try to support, and yet in HTML output all of the formatting is thrown away, so these packages are merely emulated.

Packages with graphical output are allowed as-is, but must be nested inside a `lateximage` environment to preserve the graphics.

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

Table 7: Section depths and HTML headings

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

* If `FormatWP` is true, section headings may be adjusted, depending on `WPTitleHeading`. See table 6 on page 116.

14 Section depths and HTML headings

Stacks are created to track depth inside the \LaTeX document structure. This depth is translated to HTML headings as shown in table 7. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the \LaTeX document structure.

When starting a new section, the program first must close out any existing sections and lists of a deeper level to keep the HTML tags nested correctly.

Support for the memoir package will require the addition of a book level, which may push the HTML headings down a step, and also cause subsubsection to become a <div> due to a limit of six HTML headings.

It is possible to use HTML5 <section> and <h1> for all levels, but this may not be well-recognized by older browsers.

Fixed levels for parts and chapters allow the css to remain fixed as well.

15 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:
for PRINT output:
for HTML & PRINT: Green-colored tags in the left margin show which sections of source code apply to the generation of HTML, print, or both forms of output.

16 Detecting the TeX Engine — pdf \LaTeX , lua \LaTeX , xe \LaTeX

```

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

```

17 pdf \LaTeX T1 and UTF8 encoding

When using pdf \LaTeX , lwarp required T1 and UTF8 encoding.

X \LaTeX and Lua \LaTeX are both UTF8 by nature.

```

6 \ifPDFTeX
7 \RequirePackage[T1]{fontenc}
8 \RequirePackage[utf8]{inputenc}
9 \fi

```

18 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.

```

10
11 \RequirePackage{newunicodechar}
12
13 \newunicodechar{*}{\texttimes}
14
15 \ifPDFTeX
16 \newunicodechar{ff}{ff}% the first arguments are ligatures
17 \newunicodechar{fi}{fi}
18 \newunicodechar{fl}{fl}
19 \newunicodechar{ffi}{ffi}
20 \newunicodechar{ffl}{ffl}
21 \newunicodechar{--}{--}
22 \newunicodechar{-}{-}

```

In PDF_TE_X, preserve upright quotes in verbatim text:

```
23 \RequirePackage{upquote}
24 \else
25 \fi
```

19 Early package requirements

- Pkg `etoolbox` Provides `\ifbool` and other functions.
- Pkg `xpatch` Patches macros with optional arguments.
- 26 `\RequirePackage{etoolbox}[2011/01/03]% v2.6 for \BeforeBeginEnvironment, etc.`
27 `\RequirePackage{xpatch}`
- Pkg `ifplatform` Provides `\ifwindows` to try to automatically detect Windows OS.
- 28 `\RequirePackage{ifplatform}% sense op-system platform`
- Pkg `letltxmacro` Used to redefine `\textbf` and friends.
- 29 `\RequirePackage{letltxmacro}`

20 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	

20.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

```
30 \newbool{usingOSWindows}
31 \boolfalse{usingOSWindows}
```

20.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

20.3 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.

```
33 \newcommand*{\LWR@setOSWindows}
34 {
35 \booltrue{usingOSWindows}
36 \renewcommand*{\OSPathSymbol}{\@backslashchar}
37 }
```

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

```
38 \ifwindows
39 \LWR@setOSWindows
40 \fi
```

21 Package options

Pkg `kvoptions` Allows key/value package options.

```
41 \RequirePackage{kvoptions}
42 \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:

```
43 \newbool{warpingprint}
44 \newbool{warpingHTML}
45 \newbool{mathjax}
```

[defaults](#) The default is print output, and svg math if the user chose HTML output.

```
46 \booltrue{warpingprint}%
47 \boolfalse{warpingHTML}%
48 \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.

```
49 \DeclareVoidOption{warpprint}{%
50 \PackageInfo{lwarp}{Using option 'warpprint'}
51 \booltrue{warpingprint}%
52 \boolfalse{warpingHTML}%
53 }
```

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.

```
54 \DeclareVoidOption{warpHTML}{%
55 \PackageInfo{lwarp}{Using option 'warpHTML'}%
```

```
56 \booltrue{warpingHTML}%
57 \boolfalse{warpingprint}%
58 }
```

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.

```
59 \DeclareVoidOption{mathsvg}{%
60 \PackageInfo{lwarp}{Using option 'mathsvg'}}
61 \boolfalse{mathjax}%
62 }
```

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

```
63 \DeclareVoidOption{mathjax}{%
64 \PackageInfo{lwarp}{Using option 'mathjax'}}
65 \booltrue{mathjax}%
66 }
```

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.

```
67 \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.

```
68 \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 "}")
```

```
69 \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`:

```

70 \newbool{LWR@creatinglwarpmk}
71 \boolfalse{LWR@creatinglwarpmk}
72
73 \DeclareVoidOption{lwarpmk}{
74 \PackageInfo{lwarpmk}{Using option 'lwarpmk'}}
75 \booltrue{LWR@creatinglwarpmk}
76 }

```

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.

```

77 \DeclareVoidOption{OSWindows}{
78 \PackageInfo{lwarpmk}{Using option 'OSWindows'}}
79 \LWR@setOSWindows
80 }

```

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.
Default: \lwarpmk

```
81 \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.
Default: <empty>

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

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

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

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

```
84 \ProcessKeyvalOptions*\relax
```

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

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

Defaults unless already over-ridden by the user:

```
86 \ifcempty{LWR@HomeHTMLFilename}{
87 \newcommand*{\HomeHTMLFilename}{\BaseJobname}
```

```

88 }{
89 \csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
90 }
91
92 \csedef{HTMLFilename}{\LWR@HTMLFilename}

```

21.1 Conditional compilation

`\warpprintonly` $\{\langle contents \rangle\}$

Only process the contents if producing printed output.

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

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

Only process the contents if producing HTML output.

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

`\Pkg` `comment` Provides conditional code blocks.

```
95 \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.

```

96 \ifbool{warpingHTML}{
97 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
98 }{}
99
100 \ifbool{warpingprint}{
101 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
102 }{}

```

```
103 \excludecomment{testing}
```

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

```
104 \includecomment{warpall}
```

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

`\Env` `warpHTML`

For HTML output:

```
105 \ifbool{warpingHTML}{%
106 \includecomment{warpHTML}
107 }
108 {\excludecomment{warpHTML}}%
```

```
109 \ifbool{warpingprint}
110 {\includecomment{warpprint}}
111 {\excludecomment{warpprint}}
```

Optionally generate a local copy of `lwarpmk`. Default to no.

```
112 \ifbool{LWR@creatinglwarpmk}
113 {\includecomment{LWR@createlwarpmk}}
114 {\excludecomment{LWR@createlwarpmk}}
```

22 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` $\langle\textit{packagename}\rangle$ Error if this package was loaded before `lwarp`.

```
115 \newcommand*\LWR@loadafter}[1]{%
116 \@ifpackageloaded{#1}
117 {
118 \PackageError{lwarp}
119 {Package #1, or one which uses #1, must be loaded after lwarp}
120 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
121 Package #1 may also be loaded by something else, which must also be moved
122 after lwarp.}
123 }
124 {}
125 }
```

`\LWR@loadbefore` $\langle\textit{packagename}\rangle$ Error if this package is after `lwarp`.

```

126 \newcommand*{\LWR@loadbefore}[1]{%
127 \@ifpackageloaded{#1}
128 {}
129 {
130 \PackageError{lwarp}
131 {Package #1 must be loaded before lwarp}
132 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
133 }
134 }

```

`\LWR@loadnever` $\langle\{badpackagename}\rangle$ $\langle\{replacementpkgname}\rangle$

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

```

135 \newcommand*{\LWR@loadnever}[2]{%
136 \PackageError{lwarp}
137 {Package #1 is not supported by lwarp's HTML conversion.
138 Package(s) #2 may be useful instead}
139 {Package #1 might conflict with lwarp in some way,
140 or is superceded by another package.
141 For a possible alternative, see package(s) #2.}
142 }

```

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:

```

143 \LWR@loadafter{a4}
144 \LWR@loadafter{a4wide}
145 \LWR@loadafter{a5comb}
146 \LWR@loadafter{abstract}
147 \LWR@loadafter{adjmulticol}
148 \LWR@loadafter{addlines}
149 \LWR@loadafter{afterpage}
150 \LWR@loadafter{algorithmicx}
151 \LWR@loadafter{alltt}
152 \LWR@loadafter{amsmath}
153 \LWR@loadafter{amsthm}
154 \LWR@loadafter{anonchap}
155 \LWR@loadafter{anysize}
156 \LWR@loadafter{appendix}
157 \LWR@loadafter{arabicfront}
158 \LWR@loadafter{array}
159 \LWR@loadafter{atbegshi}
160 \LWR@loadafter{authblk}
161 \LWR@loadafter{balance}
162 \LWR@loadafter{bigdelim}

```

```
163 \LWR@loadafter{bigstrut}
164 \LWR@loadafter{bookmark}
165 \LWR@loadafter{booktabs}
166 \LWR@loadafter{boxedminipage}
167 \LWR@loadafter{boxedminipage2e}
168 \LWR@loadafter{breakurl}
169 \LWR@loadafter{cancel}
170 \LWR@loadafter{ccaption}
171 \LWR@loadafter{changepage}
172 \LWR@loadafter{chnpage}
173 \LWR@loadafter{chappg}
174 \LWR@loadafter{color}
175 \LWR@loadafter{crop}
176 \LWR@loadafter{cuted}
177 \LWR@loadafter{cutwin}
178 \LWR@loadafter{dblfnote}
179 \LWR@loadafter{dcolumn}
180 \LWR@loadafter{draftwatermark}
181 \LWR@loadafter{easy-todo}
182 \LWR@loadafter{ebook}
183 \LWR@loadafter{ellipsis}
184 \LWR@loadafter{emptypage}
185 \LWR@loadafter{enumerate}
186 \LWR@loadafter{enumitem}
187 \LWR@loadafter{epigraph}
188 \LWR@loadafter{eso-pic}
189 \LWR@loadafter{everypage}
190 \LWR@loadafter{everyshi}
191 \LWR@loadafter{extramarks}
192 \LWR@loadafter{fancybox}
193 \LWR@loadafter{fancyhdr}
194 \LWR@loadafter{fancyref}
195 \LWR@loadafter{fancyvrb}
196 \LWR@loadafter{figcaps}
197 \LWR@loadafter{fix2col}
198 \LWR@loadafter{fixme}
199 \LWR@loadafter{fixmetodonotes}
200 \LWR@loadafter{float}
201 \LWR@loadafter{floatflt}
202 \LWR@loadafter{floatpag}
203 \LWR@loadafter{floatrow}
204 \LWR@loadafter{flushend}
205 \LWR@loadafter{fncychap}
206 \LWR@loadafter{fnpos}
207 % fontenc must be loaded before lwarp
208 % fontspec must be loaded before lwarp
209 \LWR@loadafter{ftnright}
210 \LWR@loadafter{fullpage}
211 \LWR@loadafter{fullwidth}
212 \LWR@loadafter{geometry}
```

```
213 \LWR@loadafter{glossaries}
214 % \LWR@loadafter{graphics}% pre-loaded by xunicode
215 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
216 \LWR@loadafter{grffile}
217 \LWR@loadafter{hang}
218 \LWR@loadafter{hyperref}
219 \LWR@loadafter{hyperxmp}
220 \LWR@loadafter{idxlayout}
221 \LWR@loadafter{ifoddpape}
222 \LWR@loadafter{indentfirst}
223 % inputenc must be loaded before lwarp
224 \LWR@loadafter{keyfloat}
225 \LWR@loadafter{layout}
226 \LWR@loadafter{letterspace}
227 \LWR@loadafter{letrine}
228 \LWR@loadafter{lips}
229 \LWR@loadafter{listings}
230 \LWR@loadafter{longtable}
231 \LWR@loadafter{lscap}
232 \LWR@loadafter{ltcaption}
233 \LWR@loadafter{ltxtable}
234 \LWR@loadafter{luatodonotes}
235 \LWR@loadafter{marginfit}
236 \LWR@loadafter{marginfix}
237 \LWR@loadafter{marginnote}
238 \LWR@loadafter{mcaption}
239 \LWR@loadafter{mdframed}
240 \LWR@loadafter{metalogo}
241 \LWR@loadafter{microtype}
242 \LWR@loadafter{midfloat}
243 \LWR@loadafter{moreverb}
244 % morewrites must be loaded before lwarp
245 \LWR@loadafter{mparhack}
246 % \LWR@loadafter{multicol}% loaded by ltxdoc
247 \LWR@loadafter{multirow}
248 \LWR@loadafter{nameref}
249 \LWR@loadafter{needspace}
250 % newclude must be loaded before lwarp
251 \LWR@loadafter{newtxmath}
252 % newunicodechar must be loaded before lwarp
253 \LWR@loadafter{nextpage}
254 \LWR@loadafter{nonumonpart}
255 \LWR@loadafter{nopageno}
256 \LWR@loadafter{nowidow}
257 \LWR@loadafter{ntheorem}
258 \LWR@loadafter{overpic}
259 \LWR@loadafter{pagenote}
260 \LWR@loadafter{paralist}
261 \LWR@loadafter{parskip}
262 \LWR@loadafter{pdfscape}
```

```
263 \LWR@loadafter{pdfsync}
264 \LWR@loadafter{pfnote}
265 \LWR@loadafter{placeins}
266 \LWR@loadafter{prelim2e}
267 \LWR@loadafter{preview}
268 \LWR@loadafter{quotchap}
269 \LWR@loadafter{ragged2e}
270 \LWR@loadafter{realscripts}
271 \LWR@loadafter{relsize}
272 \LWR@loadafter{romanbar}
273 \LWR@loadafter{romanbarpagenumber}
274 \LWR@loadafter{rotating}
275 \LWR@loadafter{rotfloat}
276 \LWR@loadafter{savetrees}
277 % \LWR@loadafter{scalefnt}% loaded by babel-french
278 \LWR@loadafter{scrextend}
279 \LWR@loadafter{scrhack}
280 \LWR@loadafter{scllayer}
281 \LWR@loadafter{scllayer-notecolumn}
282 \LWR@loadafter{scllayer-scrpage}
283 \LWR@loadafter{section}
284 \LWR@loadafter{sectsty}
285 \LWR@loadafter{setspace}
286 \LWR@loadafter{shadow}
287 \LWR@loadafter{showidx}
288 \LWR@loadafter{showkeys}
289 \LWR@loadafter{sidecap}
290 \LWR@loadafter{sidenotes}
291 \LWR@loadafter{siunitx}
292 \LWR@loadafter{soul}
293 \LWR@loadafter{soulpos}
294 \LWR@loadafter{soulutf8}
295 \LWR@loadafter{stabular}
296 \LWR@loadafter{subfig}
297 \LWR@loadafter{supertabular}
298 \LWR@loadafter{tabls}
299 \LWR@loadafter{tabularx}
300 \LWR@loadafter{tabulary}
301 \LWR@loadafter{textarea}
302 % \LWR@loadafter{textcomp}% maybe before lwarp with font packages
303 \LWR@loadafter{textpos}
304 \LWR@loadafter{theorem}
305 \LWR@loadafter{threeparttable}
306 \LWR@loadafter{tikz}
307 \LWR@loadafter{titleps}
308 \LWR@loadafter{titlesec}
309 \LWR@loadafter{titletoc}
310 \LWR@loadafter{titling}
311 % \LWR@loadafter{tocbasic}% preloaded by koma-script classes
312 \LWR@loadafter{tocbibind}
```

```

313 \LWR@loadafter{tocloft}
314 \LWR@loadafter{tocstyle}
315 \LWR@loadafter{todo}
316 \LWR@loadafter{todonotes}
317 \LWR@loadafter{transparent}
318 \LWR@loadafter{trivfloat}

319 % \LWR@loadafter{typearea}% preloaded by koma-script classes
320 \LWR@loadafter{ulem}
321 \LWR@loadafter{upref}
322 \LWR@loadafter{varioref}
323 \LWR@loadafter{verse}
324 \LWR@loadafter{wallpaper}
325 \LWR@loadafter{watermark}
326 \LWR@loadafter{wrapfig}
327 \LWR@loadafter{xcolor}
328 \LWR@loadafter{xfrac}
329 \LWR@loadafter{xltextra}
330 \LWR@loadafter{xmpincl}
331 \LWR@loadafter{xtab}
332 \LWR@loadafter{zwpagelayout}

```

23 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: `333 \begin{warpall}`

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

Detects $X_{\text{}}\text{T}_{\text{}}\text{E}_{\text{}}\text{X}$ and $\text{L}\text{u}\text{a}\text{T}_{\text{}}\text{E}_{\text{}}\text{X}$:

```

334 \RequirePackage{iftex}
335 \newif\ifxetexorluatex
336 \ifXeTeX
337     \xetexorluatextrue
338 \else
339     \ifLuaTeX
340         \xetexorluatextrue

```

```

341 \else
342     \xetexorluatexfalse
343 \fi
344 \fi

345 \end{warpall}

```

for HTML output: 346 \begin{warpHTML}

```

347 \ifxetexorluatex
348 % ^^A \usepackage[no-math]{fontspec}

```

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

```

349 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
350 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
351 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
352 \else

```

pdf_latex only: Only pre-loaded if pdf_latex 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.

```

353 \RequirePackage {microtype}
354
355 \microtypesetup{
356     protrusion=false,
357     expansion=false,
358     tracking=false,
359     kerning=false,
360     spacing=false}
361
362 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}

363 \fi

364 \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:

```

365 \begin{warpHTML}
366 \RequirePackage[paperheight=190in,paperwidth=20in,%
367 left=2in,right=12in,%
368 top=1in,bottom=1in,%
369 ]{geometry}
370 \@twosidefalse
371 \@mparswitchfalse
372 \end{warpHTML}

```

for HTML & PRINT:

```

373 \begin{warpall}

```

Pkg `xparse`

L^AT_EX3 command argument parsing

```

374 \RequirePackage{xparse}

```

Pkg `afterpackage` Used to patch titling to add `\AddSubTitlePublished`. Provided by `scrfile` or `afterpackage`.

```

375 \@ifundefined{AfterPackage}%
376 {
377 \RequirePackage{afterpackage}
378 }{}

```

```

379 \end{warpall}

```

for HTML output:

```

380 \begin{warpHTML}

```

Pkg `expl3`

L^AT_EX3 programming

```

381 \RequirePackage{expl3}

```

Pkg `getttitlestring`

Used to emulate `\nameref`.

```

382 \RequirePackage{getttitlestring}

```

Pkg `everyhook`

everyhook is used to patch paragraph handling.

```
383 \RequirePackage{everyhook}
384 \end{warpHTML}
```

for HTML & PRINT: 385 \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>

```
386 \RequirePackage{filecontents}
387
388 \@ifpackagelater{filecontents}{2011/10/09}%
389 {}
390 {
391 \newwrite\fcwrite
392 \let\LWR@origfilec@ntents\filec@ntents
393 \def\filec@ntents{\def\chardef##1\write{\let\reservedc\fcwrite}\LWR@origfilec@ntents}
394 }
```

```
395 \end{warpall}
```

for HTML output: 396 \begin{warpHTML}

Pkg xifthen

```
397 \RequirePackage{xifthen}
```

Pkg xstring

```
398 \RequirePackage{xstring}
```

Pkg xstring

```
399 \RequirePackage{verbatim}
```

Pkg makeidx

```
400 \RequirePackage{makeidx}
401 \makeindex
```

Pkg calc

```
402 \RequirePackage{calc}
```

Pkg `refcount`

```
403 \RequirePackage{refcount}
```

Pkg `newfloat`

```
404 \RequirePackage{newfloat}
```

Pkg `caption`

```
405 \RequirePackage{caption}
```

```
406 \end{warpHTML}
```

for HTML & PRINT: `407 \begin{warpall}`

Pkg `environ`

Used to encapsulate math environments for re-use in HTML `<alt>` text.

```
408 \RequirePackage{environ}
```

```
409 \end{warpall}
```

for HTML output: `410 \begin{warpHTML}`

Pkg `zref`

Used for cross-references.

```
411 \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.

```
412 \PassOptionsToPackage{leqno}{amsmath}
```

```
413 \RequirePackage{amsmath}
```

Used to convert lengths for image width/height options.

```
414 \RequirePackage{printlen}
```

```
415 \end{warpHTML}
```

24 Loading packages

for HTML output: 416 `\begin{warpHTML}`

Remember the original `\RequirePackage`:

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

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

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

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

Note that argument 4 is passed directly to `\StrBetween`.

```
419 \newcommand*\LWR@findword[3][,]{%
420   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
421 }
```

`\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.

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

Find the *n*'th package name from the list:

```
423 \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`.

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

See if the package name was found:

```
425 \IfStrEq{\LWR@strresulttwo}{}%
426 {}% no filename
427 {}% yes filename
```

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

```
428 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
429 {}% lwarp-* file found
430   \StrSubstitute%
```

```

431         {\LWR@requirepackagenames}%
432         {\LWR@strresult}%
433         {lwarp-\LWR@strresulttwo}[\LWR@requirepackagenames]%
434 }%
435 {}% no lwarp-* file
436 }% yes filename
437 }

```

`\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 \TeX version.

```

438 \RenewDocumentCommand{\RequirePackage}{o m o}{%

```

Redirect up to nine names:

```

439 \renewcommand*{\LWR@requirepackagenames}{#2}
440 \LWR@lookforpackagename{1}
441 \LWR@lookforpackagename{2}
442 \LWR@lookforpackagename{3}
443 \LWR@lookforpackagename{4}
444 \LWR@lookforpackagename{5}
445 \LWR@lookforpackagename{6}
446 \LWR@lookforpackagename{7}
447 \LWR@lookforpackagename{8}
448 \LWR@lookforpackagename{9}

```

`\RequirePackage` depending on the options and version:

```

449 \IfValueTF{#1}
450 {% options given
451     \IfValueTF{#3}% version given?
452     {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}[#3]}
453     {\LWR@origRequirePackage[#1]{\LWR@requirepackagenames}}
454 }
455 {% no options given
456     \IfValueTF{#3}% version given?
457     {\LWR@origRequirePackage{\LWR@requirepackagenames}[#3]}
458     {\LWR@origRequirePackage{\LWR@requirepackagenames}}
459 }
460 }
461 \LetLtxMacro{\usepackage}{\RequirePackage}

```

`\LWR@ProvidesPackagePass` {*⟨pkgname⟩*} [*⟨version⟩*]

Uses the original package, including options.

```

462 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
463 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
464 \IfValueTF{#2}
465 {\ProvidesPackage{lwarp-#1}[#2]}
466 {\ProvidesPackage{lwarp-#1}}
467 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
468 \ProcessOptions\relax
469
470 \IfValueTF{#2}
471 {\LWR@origRequirePackage{#1}[#2]}
472 {\LWR@origRequirePackage{#1}}
473 }

```

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

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

```

474 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
475 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
476 \IfValueTF{#2}
477 {\ProvidesPackage{lwarp-#1}[#2]}
478 {\ProvidesPackage{lwarp-#1}}
479 \DeclareOption*{}
480 \ProcessOptions\relax
481 }

482 \end{warpHTML}

```

25 File handles

Defines file handles for writes.

for HTML & PRINT: 483 `\begin{warpall}`

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

```
484 \newwrite\LWR@quickfile%
```

```
485 \end{warpall}
```

for HTML output: 486 `\begin{warpHTML}`

`\LWR@lateximagesfile` For `lateximages.txt`.

```
487 \newwrite\LWR@lateximagesfile
```

```
488 \end{warpHTML}
```

26 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 \TeX .

for HTML output: 489 `\begin{warpHTML}`

```
\include {<filename>}
```

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

```
490 \def\@include#1 {%
491 \immediate\openout\LWR@quickfile #1_html_inc.tex% new
492 \immediate\write\LWR@quickfile{\string\input{#1.tex}}% new
493 \immediate\closeout\LWR@quickfile% new
494 \LWR@origclearpage% \changed
495 \if@filesw
496   \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
497 \fi
498 \@tempwattrue
499 \if@partsw
500   \@tempwafalse
501   \edef\reserved@b{#1}%
502   \@for\reserved@a:=\@partlist\do
503     {\ifx\reserved@a\reserved@b\@tempwattrue\fi}%
504 \fi
505 \if@tempswa
506   \let\@auxout\@partaux
507   \if@filesw
508     \immediate\openout\@partaux #1_html_inc.aux % changed
509     \immediate\write\@partaux{\relax}%
510   \fi
511   \@input{#1_html_inc.tex}% changed
```

```

512 \LWR@origclearpage% changed
513 \@writeckpt{#1}%
514 \if@filesw
515     \immediate\closeout\@partaux
516 \fi
517 \else
518     \deadcycles\z@
519     \@nameuse{cp@#1}%
520 \fi
521 \let\@auxout\@mainaux%
522 }

523 \end{warpHTML}

```

27 Copying a file

for HTML output: 524 \begin{warpHTML}

\LWR@copyfile *{(source filename)}* *{(destination filename)}*

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

```

525 \newwrite\LWR@copyoutfile % open the file to write to
526 \newread\LWR@copyinfile % open the file to read from
527
528 \newcommand*{\LWR@copyfile}[2]{%
529 \immediate\openout\LWR@copyoutfile=#2
530 \openin\LWR@copyinfile=#1
531 \begingroup\endlinechar=-1
532 \makeatletter
533 \loop\unless\ifeof\LWR@copyinfile
534 \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
535 % \LWR@fileline\par % print the content into the pdf
536 % print the content:
537 \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
538 \repeat
539 \closeout\LWR@copyoutfile
540 \endgroup
541 }

542 \end{warpHTML}

```

28 Debugging messages

```

for HTML & PRINT: 543 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

544 \newbool{LWR@tracinglwarp}

\tracinglwarp Turns on the debug tracing messages.

545 \newcommand{\tracinglwarp}{\booltrue{LWR@tracinglwarp}}

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

546 \newcommand{\LWR@traceinfo}[1]{%
547 \ifbool{LWR@tracinglwarp}%
548 {%
549   \typeout{*** lwarp: #1}%
550   % \PackageInfo{lwarp}{#1 : }%
551 }%
552 {}%
553 }

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

554 \newbool{HTMLDebugComments}
555 \boolfalse{HTMLDebugComments}

If \tracinglwarp, show where preamble hooks occur:

556 \AfterEndPreamble{
557 \LWR@traceinfo{AfterEndPreamble}
558 }
559
560 \AtBeginDocument{
561 \LWR@traceinfo{AtBeginDocument}
562 }

563 \end{warpall}

```

29 Koma-script

Load patches to koma-script.

for HTML output: 564 `\begin{warpHTML}`

```
565 \@ifclassloaded{scrbook}{\RequirePackage{lwarp-patch-komascript}}{}
566 \@ifclassloaded{scrartcl}{\RequirePackage{lwarp-patch-komascript}}{}
567 \@ifclassloaded{scrreprt}{\RequirePackage{lwarp-patch-komascript}}{}

568 \end{warpHTML}
```

30 HTML-conversion output modifications

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

for HTML & PRINT: 569 `\begin{warpall}`

30.1 User-level controls

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

```
570 \newbool{FormatEPUB}
571 \boolfalse{FormatEPUB}
```

Bool `FormatWP` 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.
 Default: `false`

```
572 \newbool{FormatWP}
573 \boolfalse{FormatWP}
```

Bool `WPMarkFloats` Adds

Default: `false`

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

OR

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

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

```
574 \newbool{WPMarkFloats}
575 \boolfalse{WPMarkFloats}
```

Bool **WPMarkMinipages** Adds
 Default: *false* `=== begin minipage ===`
 `...`
 `=== end minipage ===`

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

```
576 \newbool{WPMarkMinipages}
577 \boolfalse{WPMarkMinipages}
```

Bool **WPMarkTOC** While formatting for word processors, adds
 Default: *true* `=== table of contents ===`

where the Table of Contents would have been. This helps identify where to insert the actual TOC.

If set false, the actual TOC is printed instead.

```
578 \newbool{WPMarkTOC}
579 \booltrue{WPMarkTOC}
```

Bool **WPMarkLOFT** While formatting for word processors, adds
 Default: *false* `=== list of figures === and/or`
 `=== list of tables ===`

where each of these lists would have been. This helps identify where to insert the actual lists.

If set false, the actual lists are printed instead.

```
580 \newbool{WPMarkLOFT}
581 \boolfalse{WPMarkLOFT}
```

Bool **WPMarkMath** While formatting for word processors, prints math as \LaTeX code instead of creat-

¹²Perhaps some day word processors will have HTML import options for identifying `<figure>` and `<figcaption>` tags for figures and tables.

ing SVG images or MathJax. This is useful for cut/paste into the LibreOffice Writer TeXMaths extension.

```
582 \newbool{WPMarkMath}
583 \boolfalse{WPMarkMath}
```

Bool WPTitleHeading While formatting for word processors, true sets the document title to `<h1>`, which is expected for HTML documents, but also causes the lower-level section headings to start at **Heading 2** when imported into LibreOffice. Set to `false` to cause the title to be plain text, and the section headings to begin at **Heading 1**.

Default: `false`

See table 6 on page 116.

```
584 \newbool{WPTitleHeading}
585 \boolfalse{WPTitleHeading}
```

```
586 \end{warpall}
```

30.2 Heading adjustments

If formatting the HTML for a word processor, adjust heading levels.

If WPTitleHeading is true, adjust so that part is **Heading 1**.

If WPTitleHeading is false, use `<h1>` for the title, and set part to **Heading 2**.

for HTML output: `587 \begin{warpHTML}`

```
588 \AtBeginDocument{
589 \ifbool{FormatWP}{
590 \@ifundefined{chapter}{
591 \ifbool{WPTitleHeading}{% part and section starting at h2
592 \renewcommand*\LWR@tagtitle}{h1}
593 \renewcommand*\LWR@tagtitleend}{/h1}
594 \renewcommand*\LWR@tagpart}{h2}
595 \renewcommand*\LWR@tagpartend}{/h2}
596 \renewcommand*\LWR@tagsection}{h3}
597 \renewcommand*\LWR@tagsectionend}{/h3}
598 \renewcommand*\LWR@tagsubsection}{h4}
599 \renewcommand*\LWR@tagsubsectionend}{/h4}
600 \renewcommand*\LWR@tagsubsubsection}{h5}
601 \renewcommand*\LWR@tagsubsubsectionend}{/h5}
602 \renewcommand*\LWR@tagparagraph}{h6}
603 \renewcommand*\LWR@tagparagraphend}{/h6}
604 \renewcommand*\LWR@tagsubparagraph}{span class="subparagraph"{} }
605 \renewcommand*\LWR@tagsubparagraphend}{/span}
606 }% WPTitleHeading
```

```

607 {% not WPTitleHeading, part and section starting at h1
608 \renewcommand*\LWR@tagtitle}{div class="title"}
609 \renewcommand*\LWR@tagtitleend}{/div}
610 \renewcommand*\LWR@tagpart}{h1}
611 \renewcommand*\LWR@tagpartend}{/h1}
612 \renewcommand*\LWR@tagsection}{h2}
613 \renewcommand*\LWR@tagsectionend}{/h2}
614 \renewcommand*\LWR@tagsubsection}{h3}
615 \renewcommand*\LWR@tagsubsectionend}{/h3}
616 \renewcommand*\LWR@tagsubsubsection}{h4}
617 \renewcommand*\LWR@tagsubsubsectionend}{/h4}
618 \renewcommand*\LWR@tagparagraph}{h5}
619 \renewcommand*\LWR@tagparagraphend}{/h5}
620 \renewcommand*\LWR@tagsubparagraph}{h6}
621 \renewcommand*\LWR@tagsubparagraphend}{/h6}
622 }% not WPTitleHeading
623 }% chapter undefined
624 {% chapter defined
625 \ifbool{WPTitleHeading}{}
626 {% not WPTitleHeading, part and chapter starting at h1
627 \renewcommand*\LWR@tagtitle}{div class="title"}
628 \renewcommand*\LWR@tagtitleend}{/div}
629 \renewcommand*\LWR@tagpart}{h1}
630 \renewcommand*\LWR@tagpartend}{/h1}
631 \renewcommand*\LWR@tagchapter}{h2}
632 \renewcommand*\LWR@tagchapterend}{/h2}
633 \renewcommand*\LWR@tagsection}{h3}
634 \renewcommand*\LWR@tagsectionend}{/h3}
635 \renewcommand*\LWR@tagsubsection}{h4}
636 \renewcommand*\LWR@tagsubsectionend}{/h4}
637 \renewcommand*\LWR@tagsubsubsection}{h5}
638 \renewcommand*\LWR@tagsubsubsectionend}{/h5}
639 \renewcommand*\LWR@tagparagraph}{h6}
640 \renewcommand*\LWR@tagparagraphend}{/h6}
641 \renewcommand*\LWR@tagsubparagraph}{span class="subparagraph"}}}
642 \renewcommand*\LWR@tagsubparagraphend}{/span}
643 }% not WPTitleHeading
644 }% chapter defined
645 }{}% FormatWP
646 }% AtBeginDocument

647 \end{warpHTML}

```

31 Remembering original formatting macros

for HTML output: 648 \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.

```

649 \LetLtxMacro{\LWR@origtextrm}{\textrm}
650 \LetLtxMacro{\LWR@origtextsf}{\textsf}
651 \LetLtxMacro{\LWR@origtexttt}{\texttt}
652 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
653 \LetLtxMacro{\LWR@origtextbf}{\textbf}
654 \LetLtxMacro{\LWR@origtextmd}{\textmd}
655 \LetLtxMacro{\LWR@origtextit}{\textit}
656 \LetLtxMacro{\LWR@origtextsl}{\textsl}
657 \LetLtxMacro{\LWR@origtextsc}{\textsc}
658 \LetLtxMacro{\LWR@origtextup}{\textup}
659 \LetLtxMacro{\LWR@origemph}{\emph}
660
661 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
662 \LetLtxMacro{\LWR@origsfamily}{\sffamily}
663 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
664 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
665 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
666 \LetLtxMacro{\LWR@origupshape}{\upshape}
667 \LetLtxMacro{\LWR@origslshape}{\slshape}
668 \LetLtxMacro{\LWR@origscshape}{\scshape}
669 \LetLtxMacro{\LWR@origitshape}{\itshape}
670 \LetLtxMacro{\LWR@origem}{\em}
671 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
672
673 \let\LWR@origraggedright\raggedright
674 \let\LWR@origonecolumn\onecolumn
675
676 \let\LWR@origsp\sp
677 \let\LWR@origsb\sb
678 \LetLtxMacro\LWR@origtextsuperscript\textsuperscript
679 \LetLtxMacro\LWR@orig@textsuperscript@\textsuperscript
680 \AtBeginDocument{
681 \LetLtxMacro\LWR@origtextsubscript\textsubscript
682 \LetLtxMacro\LWR@orig@textsubscript@\textsubscript
683 }
684 \LetLtxMacro\LWR@origunderline\underline
685
686 \let\LWR@origscriptsize\scriptsize
687
688 \let\LWR@orignewpage\newpage
689
690 \let\LWR@origpagestyle\pagestyle
691 \let\LWR@origthispagestyle\thispagestyle
692 \let\LWR@origpagenumbering\pagenumbering
693

```

```

694 \LetLtxMacro{\LWR@origminipage}{\minipage}
695 \let\LWR@origendminipage\endminipage
696 \LetLtxMacro{\LWR@origparbox}{\parbox}
697
698 \let\LWR@orignewline\newline
699
700 \LetLtxMacro\LWR@origitem\item
701
702
703 \AtBeginDocument{% in case packages change definition
704 \LetLtxMacro{\LWR@origitemize}{\itemize}
705 \LetLtxMacro{\LWR@endorigitemize}{\enditemize}
706 \LetLtxMacro{\LWR@origenumerate}{\enumerate}
707 \LetLtxMacro{\LWR@endorigenumerate}{\endenumerate}
708 \LetLtxMacro{\LWR@origdescription}{\description}
709 \LetLtxMacro{\LWR@endorigdescription}{\enddescription}
710 }
711
712 \let\LWR@origpar\par
713
714 \LetLtxMacro{\LWR@origfootnote}{\footnote}
715 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
716
717 \let\LWR@origclearpage\clearpage

718 \end{warpHTML}

```

32 Accents

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

for HTML output: 719 `\begin{warpHTML}`

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

```
720 \AtBeginDocument{
```

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

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

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

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

```

722 \ifPDFTeX
723 \LetLtxMacro{\LWR@origequalaccent}{\=}
724 \LetLtxMacro{\LWR@origdotaccent}{\cdot}
725 \LetLtxMacro{\LWR@origu}{\u}
726 \LetLtxMacro{\LWR@origv}{\v}
727 \LetLtxMacro{\LWR@origc}{\c}
728 \LetLtxMacro{\LWR@origd}{\d}
729 \LetLtxMacro{\LWR@origb}{\b}

```

The HTML redefinitions follow.

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

```

730 \renewcommand*{\=} [1]{#1\HTMLUnicode{0305}}
731 \renewcommand*{\cdot} [1]{#1\HTMLUnicode{0307}}
732 \renewcommand*{\u} [1]{#1\HTMLUnicode{0306}}
733 \renewcommand*{\v} [1]{#1\HTMLUnicode{030C}}
734 \renewcommand*{\c} [1]{#1\HTMLUnicode{0327}}
735 \renewcommand*{\d} [1]{#1\HTMLUnicode{0323}}
736 \renewcommand*{\b} [1]{#1\HTMLUnicode{0331}}
737 \fi

```

For all engines, a Unicode diacritical tie is used:

```

738 \def\LWR@t#1#2{#1\HTMLUnicode{0361}#2}
739 \renewcommand*{\t} [1]{\LWR@t#1}

```

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

```

740 \ifPDFTeX
741 \newcommand*{\LWR@restoreorigaccents}{%
742 \LetLtxMacro{\=}{\LWR@origequalaccent}%
743 \LetLtxMacro{\cdot}{\LWR@origdotaccent}%
744 \LetLtxMacro{\u}{\LWR@origu}%
745 \LetLtxMacro{\v}{\LWR@origv}%
746 \LetLtxMacro{\t}{\LWR@origt}%
747 \LetLtxMacro{\c}{\LWR@origc}%
748 \LetLtxMacro{\d}{\LWR@origd}%
749 \LetLtxMacro{\b}{\LWR@origb}%
750 }
751 \else% XeLaTeX, LuaLaTeX:
752 \newcommand*{\LWR@restoreorigaccents}{%
753 \LetLtxMacro{\t}{\LWR@origt}%
754 }
755 \fi
756 }% AtBeginDocument

757 \end{warpHTML}

```

33 Configuration Files

```
758 \begin{warpprint}
759 \typeout{lwarp: generating configuration files}
760 \end{warpprint}
```

33.1 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.

```
761 \begin{warpprint}
762 \immediate\openout\LWR@quickfile=\jobname_html.tex
763 \immediate\write\LWR@quickfile{%
764 \detokenize{\PassOptionsToPackage}%
765 {warpHTML,BaseJobname=\jobname}{lwarp}%
766 }
767 \immediate\write\LWR@quickfile{%
768 \detokenize{\input}\string{\jobname.tex}\string }%
769 }
770 \immediate\closeout\LWR@quickfile
771 \end{warpprint}
```

33.2 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:

```
772 \begin{warpprint}
773 \ifcsdef\LWR@quickfile-{}{\newwrite\LWR@quickfile}
774 \immediate\openout\LWR@quickfile=lwarpmk.conf
775 \ifbool{usingOSWindows}{
```

```

776 \immediate\write\LWR@quickfile{opsystem = "Windows"}
777 }{
778 \immediate\write\LWR@quickfile{opsystem = "Unix"}
779 }
780 \ifPDFTeX
781 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
782 \fi
783 \ifXeTeX
784 \immediate\write\LWR@quickfile{latexname = "xelatex"}
785 \fi
786 \ifLuaTeX
787 \immediate\write\LWR@quickfile{latexname = "lualatex"}
788 \fi
789 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
790 \immediate\write\LWR@quickfile{%
791 homehtmlfilename = "\HomeHTMLFilename"%
792 }
793 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
794 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
795 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
796 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
797 \immediate\closeout\LWR@quickfile
798 \end{warpprint}

```

33.3 project.lwarpmkconf

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

```

799 \begin{warpprint}
800 \ifcsdef{LWR@quickfile}{}{\newwrite{\LWR@quickfile}}
801 \immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
802 \ifbool{usingOSWindows}{
803 \immediate\write\LWR@quickfile{opsystem = "Windows"}
804 }{
805 \immediate\write\LWR@quickfile{opsystem = "Unix"}
806 }
807 \ifPDFTeX
808 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
809 \fi
810 \ifXeTeX
811 \immediate\write\LWR@quickfile{latexname = "xelatex"}
812 \fi
813 \ifLuaTeX
814 \immediate\write\LWR@quickfile{latexname = "lualatex"}
815 \fi
816 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
817 \immediate\write\LWR@quickfile{%

```

```

818 homehtmlfilename = "\HomeHTMLFilename"%
819 }
820 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
821 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
822 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
823 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
824 \immediate\closeout\LWR@quickfile
825 \end{warpprint}

```

33.4 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.

```

826 \begin{warpprint}
827 \begin{filecontents*}{lwarp.css}
828 /*
829 CSS stylesheet for the LaTeX lwarp package
830 Copyright 2016-2017 Brian Dunn -- BD Tech Concepts LLC
831 */
832
833
834 /* a fix for older browsers: */
835 header, section, footer, aside, nav, main,
836     article, figure { display: block; }
837
838
839 A:link {color:#000080 ; text-decoration: none ; }
840 A:visited {color:#800000 ; }
841 A:hover {color:#000080 ; text-decoration: underline ;}
842 A:active {color:#800000 ; }
843
844 a.tocpart {display: inline-block ; margin-left: 0em ;
845     font-weight: bold ;}
846 a.tocchapter {display: inline-block ; margin-left: 0em ;
847     font-weight: bold ;}
848 a.tocsection {display: inline-block ; margin-left: 1em ;
849     text-indent: -.5em ; font-weight: bold ; }
850 a.tocsubsection {display: inline-block ; margin-left: 2em ;
851     text-indent: -.5em ; }
852 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
853     text-indent: -.5em ; }
854 a.tocparagraph {display: inline-block ; margin-left: 4em ;
855     text-indent: -.5em ; }
856 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;

```

```
857     text-indent: -.5em ; }
858 a.tocfigure {margin-left: 0em}
859 a.tocsubfigure {margin-left: 2em}
860 a.toctable {margin-left: 0em}
861 a.tocsubtable {margin-left: 2em}
862 a.toctheorem {margin-left: 0em}
863 a.toclstlisting {margin-left: 0em}
864
865 body {
866     font-family: "DejaVu Serif", "Bitstream Vera Serif",
867         "Lucida Bright", Georgia, serif;
868     background: #FAF7F4 ;
869     color: black ;
870     margin:0em ;
871     padding:0em ;
872     font-size: 100% ;
873     line-height: 1.2 ;
874 }
875
876 p {margin: 1.5ex 0em 1.5ex 0em ;}
877
878 /* Holds a section number to add space between it and the name */
879 span.sectionnumber { margin-right: 0em }
880
881 /* Inserted in front of index lines */
882 span.indexitem {margin-left: 0em}
883 span.indexsubitem {margin-left: 2em}
884 span.indexsubsubitem {margin-left: 4em}
885
886 div.hidden, span.hidden { display: none ; }
887
888 kbd {
889     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
890         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
891         "Courier New", monospace;
892     font-size: 100% ;
893 }
894
895 pre { padding: 3pt ; }
896
897 span.strong { font-weight: bold; }
898
899 span.textmd { font-weight: normal; }
900
901 span.textsc { font-variant: small-caps; }
902
903 span.textsl { font-style: oblique; }
904
905 span.textup { font-variant: normal; }
906
```

```
907 span.textrm {
908     font-family: "DejaVu Serif", "Bitstream Vera Serif",
909     "Lucida Bright", Georgia, serif;
910 }
911
912 span.textsf {
913     font-family: "DejaVu Sans", "Bitstream Vera Sans",
914     Geneva, Verdana, sans-serif ;
915 }
916
917 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
918
919 span.underline {
920     text-decoration: underline ;
921     text-decoration-skip ;
922 }
923
924
925 /* For realscripts */
926 .supsubscript {
927     display: inline-block;
928     text-align:left ;
929 }
930
931 .supsubscript sup,
932 .supsubscript sub {
933     position: relative;
934     display: block;
935     font-size: .5em;
936     line-height: 1;
937 }
938
939 .supsubscript sup {
940     top: .5em;
941 }
942
943 .supsubscript sub {
944     top: .5em;
945 }
946
947 span.attribution {
948     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
949 }
950
951 span.citetitle {
952     margin-left: 1em ; font-size: 80% ; font-style: oblique;
953 }
954
955 span.poemtitle {
956     font-size: 120% ; font-weight: bold;
```

```
957 }
958
959 pre.tabbing {
960     font-family: "Linux Libertine Mono O", "Lucida Console",
961         "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
962         "Liberation Mono", "FreeMono", "Andale Mono",
963         "Nimbus Mono L", "Courier New", monospace;
964 }
965
966 blockquote {
967     margin-left: 0px ;
968     margin-right: 0px ;
969 }
970
971 /* quotchap is for the quotchap package */
972 div.quotchap {
973     font-style: oblique ;
974     overflow-x: auto ;
975     margin-left: 2em ;
976     margin-right: 2em ;
977 }
978
979 blockquote p, div.quotchap p {
980     line-height: 1.5;
981     text-align: left ;
982     font-size: .85em ;
983     margin-left: 3em ;
984     margin-right: 3em ;
985 }
986
987 /* qauthor is for the quotchap package */
988 div.qauthor {
989     display: block ;
990     text-align: right ;
991     margin-left: auto ;
992     margin-right: 2em ;
993     font-size: 80% ;
994     font-variant: small-caps;
995 }
996
997 div.qauthor p {
998     text-align: right ;
999 }
1000
1001 blockquotation {
1002     margin-left: 0px ;
1003     margin-right: 0px ;
1004 }
1005
1006 blockquotation p {
```

```
1007 line-height: 1.5;
1008 text-align: left ;
1009 font-size: .85em ;
1010 margin-left: 3em ;
1011 margin-right: 3em ;
1012 }
1013
1014 div.epigraph, div.dictum {
1015 line-height: 1.2;
1016 text-align: left ;
1017 padding: 3ex 1em 0ex 1em ;
1018 /* margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
1019 margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
1020 /* margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
1021 font-size: .85em ;
1022 max-width: 27em ;
1023 }
1024
1025
1026
1027 div.epigraphsource, div.dictumauthor {
1028 text-align:right ;
1029 margin-left:auto ;
1030 /* max-width: 50% ; */
1031 border-top: 1px solid #A0A0A0 ;
1032 padding-bottom: 3ex ;
1033 line-height: 1.2;
1034 }
1035
1036 div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
1037 div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1038 div.dictumauthor { font-style:italic }
1039
1040
1041 /* lettrine package: */
1042 span.lettrine { font-size: 3ex ; float: left ; }
1043 span.lettrinetext { font-variant: small-caps ; }
1044
1045 /* ulem and soul packages: */
1046 span.uline {
1047 text-decoration: underline ;
1048 text-decoration-skip ;
1049 }
1050
1051 span.uuline {
1052 text-decoration: underline ;
1053 text-decoration-skip ;
1054 text-decoration-style: double ;
1055 }
1056
```

```
1057 span.uwave {
1058     text-decoration: underline ;
1059     text-decoration-skip ;
1060     text-decoration-style: wavy ;
1061 }
1062
1063 span.sout {
1064     text-decoration: line-through ;
1065 }
1066
1067 span.xout {
1068     text-decoration: line-through ;
1069 }
1070
1071 span.dashuline {
1072     text-decoration: underline ;
1073     text-decoration-skip ;
1074     text-decoration-style: dashed ;
1075 }
1076
1077 span.dotuline {
1078     text-decoration: underline ;
1079     text-decoration-skip ;
1080     text-decoration-style: dotted ;
1081 }
1082
1083 span.letterspacing { letter-spacing: .2ex ; }
1084
1085 span.capsspacing {
1086     font-variant: small-caps ;
1087     letter-spacing: .1ex ;
1088 }
1089
1090 span.highlight { background: #F8E800 ; }
1091
1092
1093
1094
1095 html body {
1096     margin: 0 ;
1097     line-height: 1.2;
1098 }
1099
1100
1101 body div {
1102     margin: 0ex;
1103 }
1104
1105
1106 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
```

```
1107 {
1108     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1109         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1110         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1111         "Times New Roman", serif;
1112     font-style: normal ;
1113     font-weight: bold ;
1114     text-align: left ;
1115 }
1116
1117 h1 { /* title of the entire website, used on each page */
1118     text-align: center ;
1119     font-size: 2.5em ;
1120     padding: .4ex 0em 0ex 0em ;
1121 }
1122 h2 { font-size: 2.25em }
1123 h3 { font-size: 2em }
1124 h4 { font-size: 1.75em }
1125 h5 { font-size: 1.5em }
1126 h6 { font-size: 1.25em }
1127 span.paragraph {font-size: 1em ; font-variant: normal ;
1128     margin-right: 1em ; }
1129 span.subparagraph {font-size: 1em ; font-variant: normal ;
1130     margin-right: 1em ; }
1131
1132 div.minisec {
1133     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1134         Geneva, Verdana, sans-serif ;
1135     font-style: normal ;
1136     font-weight: bold ;
1137     text-align: left ;
1138 }
1139
1140 /* Title of the file */
1141 h1 {
1142     margin: 0ex 0em 0ex 0em ;
1143     line-height: 1.3;
1144     text-align: center ;
1145 }
1146
1147 /* Part */
1148 h2 {
1149     margin: 1ex 0em 1ex 0em ;
1150     line-height: 1.3;
1151     text-align: center ;
1152 }
1153
1154 /* Chapter */
1155 h3 {
1156     margin: 3ex 0em 1ex 0em ;
```

```
1157 line-height: 1.3;
1158 }
1159
1160 /* Section */
1161 h4 {
1162     margin: 3ex 0em 1ex 0em ;
1163     line-height: 1.3;
1164 }
1165
1166 /* Sub-Section */
1167 h5 {
1168     margin: 3ex 0em 1ex 0em ;
1169     line-height: 1.3;
1170 }
1171
1172 /* Sub-Sub-Section */
1173 h6 {
1174     margin: 3ex 0em 1ex 0em ;
1175     line-height: 1.3;
1176 }
1177
1178
1179 div.titlepage {
1180     text-align: center ;
1181 }
1182
1183 .footnotes {
1184     font-size: .85em ;
1185     margin: 3ex 1em 0ex 1em ;
1186     padding-bottom: 1ex ;
1187     border-top: 1px solid silver ;
1188 }
1189
1190 .marginpar, .marginparblock {
1191     max-width:50%;
1192     float:right;
1193     text-align:left;
1194     margin: 1ex 0.5em 1ex 1em ;
1195     padding: 1ex 0.5em 1ex 0.5em ;
1196     font-size: 85% ;
1197     border-top: 1px solid silver ;
1198     border-bottom: 1px solid silver ;
1199     overflow-x: auto;
1200 }
1201
1202 .marginpar br { margin-bottom: 2ex ; }
1203
1204 div.marginblock, div.marginparblock {
1205     max-width:50%;
1206     float:right;
```

```
1207     text-align:left;
1208     margin: 1ex 0.5em 1ex 1em ;
1209     padding: 1ex 0.5em 1ex 0.5em ;
1210     overflow-x: auto;
1211 }
1212
1213 div.marginblock div.minipage,
1214 div.marginparblock div.minipage {
1215     display: block ;
1216     margin: 0pt auto 0pt auto ;
1217 }
1218
1219 div.marginblock div.minipage p ,
1220 div.marginparblock div.minipage p
1221     { font-size: 85%}
1222
1223 div.marginblock br ,
1224 div.marginparblock br
1225     { margin-bottom: 2ex ; }
1226
1227
1228 section.textbody div.footnotes{
1229     margin: 3ex 0em 0ex 0em ;
1230     border-bottom: 2px solid silver ;
1231 }
1232
1233 .footnoteheader {
1234     border-top: 2px solid silver ;
1235     margin-top: 3ex ;
1236     padding-top: 1ex ;
1237     font-weight: bold ;
1238 }
1239
1240 .mpfootnotes {
1241     text-align: left ;
1242     font-size: .85em ;
1243     margin-left: 1em ;
1244     border-top: 1px solid silver ;
1245 }
1246
1247 /* Remove footnote top border in the title page. */
1248 div.titlepage div.mpfootnotes {
1249     border-top: none ;
1250 }
1251
1252
1253
1254 ol {
1255     margin: 1ex 1em 1ex 0em;
1256     line-height: 1.2;
```

```
1257 }
1258
1259 ul, body dir, body menu {
1260   margin: 1ex 1em 1ex 0em;
1261   line-height: 1.2;
1262 }
1263
1264 li { margin: 0ex 0em 1ex 0em; }
1265
1266 html {
1267   margin: 0;
1268   padding: 0;
1269 }
1270
1271 .programlisting {
1272   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1273               "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1274               "Courier New", monospace;
1275   margin: 1ex 0ex 1ex 0ex ;
1276   padding: .5ex 0pt .5ex 0pt ;
1277   overflow-x: auto;
1278 }
1279
1280 section.textbody>pre.programlisting {
1281   border-top: 1px solid silver ;
1282   border-bottom: 1px solid silver ;
1283 }
1284
1285
1286 .inlineprogramlisting {
1287   font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1288               "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1289               "Courier New", monospace;
1290   overflow-x: auto;
1291 }
1292
1293 span.listinglabel {
1294   display: inline-block ;
1295   font-size: 70% ;
1296   width: 4em ;
1297   text-align: right ;
1298   margin-right: 2em ;
1299 }
1300
1301 div.abstract {
1302   margin: 2em 5% 2em 5% ;
1303   padding: 1ex 1em 1ex 1em ;
1304   /* font-weight: bold ; */
1305   font-size: 90% ;
1306   text-align: left ;
```

```
1307 }
1308
1309 div.abstract dl {line-height:1.5;}
1310 div.abstract dt {color:#304070;}
1311
1312 div.abstracttitle{
1313     font-family: "URW Classico", Optima, "Linux Biolinum O",
1314         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1315         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1316     font-weight:bold;
1317     font-size:1.25em;
1318     text-align: center ;
1319 }
1320
1321 span.abstracrunintitle{
1322     font-family: "URW Classico", Optima, "Linux Biolinum O",
1323         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1324         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1325     font-weight:bold;
1326 }
1327
1328
1329 .verbatim {
1330     overflow-x: auto ;
1331 }
1332
1333 .alltt {
1334     overflow-x: auto ;
1335 }
1336
1337
1338 .bverbatim {
1339     margin: 1ex 0pt 1ex 0pt ;
1340     padding: .5ex 0pt .5ex 0pt ;
1341     overflow-x: auto ;
1342 }
1343
1344 .lverbatim {
1345     margin: 1ex 0pt 1ex 0pt ;
1346     padding: .5ex 0pt .5ex 0pt ;
1347     overflow-x: auto ;
1348 }
1349
1350 .fancyvrb {
1351     font-size:.85em ;
1352     margin: 3ex 0pt 3ex 0pt
1353 }
1354
1355 .fancyvrblabel {
1356     font-weight:bold;
```

```
1357     text-align: center ;
1358 }
1359
1360
1361 .verse {
1362     font-family: "Linux Libertine Mono O", "Lucida Console",
1363         "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1364         "Liberation Mono", "FreeMono", "Andale Mono",
1365         "Nimbus Mono L", "Courier New", monospace;
1366     margin-left: 1em ;
1367 }
1368
1369
1370 div.singlespace { line-height: 1.2 ; }
1371 div.onehalfspace { line-height: 1.5 ; }
1372 div.doublespace { line-height: 2 ; }
1373
1374
1375 /* Word processor format output: */
1376 div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
1377 div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
1378 div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ;}
1379
1380
1381
1382
1383 /* Minipage environments, vertically aligned to top, center, bottom: */
1384 .minipage, .fminipage, .fcolorminipage {
1385     /* display: inline-block ; */
1386     /* Mini pages which follow each other will be tiled. */
1387     margin: .25em .25em .25em .25em;
1388     padding: .25em .25em .25em .25em;
1389     display: inline-flex;
1390     flex-direction: column ;
1391     overflow: auto;
1392 }
1393
1394 /* Paragraphs in the flexbox did not collapse their margins. */
1395 /* Have not yet researched this. */
1396 .minipage p {margin: .75ex 0em .75ex 0em ;}
1397
1398 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1399 .fcolorbox .minipage, .fcolorboxBlock .minipage
1400     {border: none ; background: none;}
1401
1402 .fbox, .fboxBlock { border: 1px solid black ; }
1403
1404 .fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1405 .fminipage, .fcolorminipage
1406     {display: inline-block}
```

```
1407
1408 .shadowbox, .shabox {
1409     border: 1px solid black;
1410     box-shadow: 3px 3px 3px #808080 ;
1411     border-radius: 0px ;
1412     padding: .4ex .3em .4ex .3em ;
1413     margin: 0pt .3ex 0pt .3ex ;
1414     display: inline-block ;
1415 }
1416
1417 .doublebox {
1418     border: 3px double black;
1419     border-radius: 0px ;
1420     padding: .4ex .3em .4ex .3em ;
1421     margin: 0pt .3ex 0pt .3ex ;
1422     display: inline-block ;
1423 }
1424
1425 .ovalbox, .Ovalbox {
1426     border: 1px solid black;
1427     border-radius: 1ex ;
1428     padding: .4ex .3em .4ex .3em ;
1429     margin: 0pt .3ex 0pt .3ex ;
1430     display: inline-block ;
1431 }
1432
1433 .Ovalbox { border-width: 2px ; }
1434
1435 .framebox {
1436     border: 1px solid black;
1437     border-radius: 0px ;
1438     padding: .3ex .2em 0ex .2em ;
1439     margin: 0pt .1ex 0pt .1ex ;
1440     display: inline-block ;
1441 }
1442
1443
1444 .mdframed {
1445 /*     padding: 0ex ; */
1446 /*     border: 1px solid black; */
1447 /*     border-radius: 0px ; */
1448     padding: 0ex ;
1449     margin: 3ex 5% 3ex 5% ;
1450 /*     display: inline-block ; */
1451 }
1452
1453 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1454
1455 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1456
```

```
1457 .mdframedtitle {
1458     padding: .5em ;
1459     display: block ;
1460     font-size: 130%
1461 }
1462
1463 .mdframedsubtitle {
1464     padding: 0ex .5em 0ex .5em ;
1465     display: block ;
1466     font-size: 115% ;
1467 }
1468
1469 .mdframedsubsubtitle {
1470     padding: 0ex .5em 0ex .5em ;
1471     display: block ;
1472 }
1473
1474 .mdtheorem {
1475     padding: 0ex .5em 0ex .5em ;
1476     margin: 3ex 5% 3ex 5% ;
1477 /* display: inline-block ; */
1478 }
1479
1480
1481 /* framed package */
1482 .framed, pre.boxedverbatim, fcolorbox {
1483     margin: 3ex 0em 3ex 0em ;
1484     border: 1px solid black;
1485     border-radius: 0px ;
1486     padding: .3ex 1em 0ex 1em ;
1487     display: block ;
1488 }
1489
1490 .snugframed {
1491     margin: 3ex 0em 3ex 0em ;
1492     border: 1px solid black;
1493     border-radius: 0px ;
1494     display: block ;
1495 }
1496
1497 .framedleftbar {
1498     margin: 3ex 0em 3ex 0em ;
1499     border-left: 3pt solid black;
1500     border-radius: 0px ;
1501     padding: .3ex .2em .3ex 1em ;
1502     display: block ;
1503 }
1504
1505 .framedtitle {
1506     margin: 0em ;
```

```
1507     padding: 0em ;
1508     font-size: 130%
1509 }
1510
1511 .framedtitle p { padding: .3em }
1512
1513
1514
1515 dl {
1516     margin: 1ex 2em 1ex 0em;
1517     line-height: 1.3;
1518 }
1519
1520 dl dt {
1521     margin-top: 1ex;
1522     margin-left: 1em ;
1523     font-weight: bold;
1524 }
1525
1526 dl dd p { margin-top: 0em; }
1527
1528
1529 nav {
1530     font-family: "URW Classico", Optima, "Linux Biolinum 0",
1531                 "DejaVu Sans", "Bitstream Vera Sans",
1532                 Geneva, Verdana, sans-serif ;
1533     margin-bottom: 4ex ;
1534 }
1535
1536 nav p {
1537     line-height: 1.2 ;
1538     margin-top:.5ex ;
1539     margin-bottom:.5ex;
1540     font-size: .9em ;
1541 }
1542
1543
1544
1545 img, img.hyperimage, img.borderimage {
1546     max-width: 600px;
1547     border: 1px solid silver;
1548     box-shadow: 3px 3px 3px #808080 ;
1549     padding: .5% ;
1550     margin: .5% ;
1551     background: none ;
1552 }
1553
1554 img.inlineimage{
1555     padding: 0px ;
1556     box-shadow: none ;
```

```
1557     border: none ;
1558     background: none ;
1559     margin: 0px ;
1560     display: inline-block ;
1561     border-radius: 0px ;
1562 }
1563
1564 img.logoimage{
1565     max-width: 300px ;
1566     box-shadow: 3px 3px 3px #808080 ;
1567     border: 1px solid black ;
1568     background:none ;
1569     padding:0 ;
1570     margin:.5ex ;
1571     border-radius: 10px ;
1572 }
1573
1574
1575 .section {
1576 /*
1577     To have each section float relative to each other:
1578 */
1579 /*
1580     display: block ;
1581     float: left ;
1582     position: relative ;
1583     background: white ;
1584     border: 1px solid silver ;
1585     padding: .5em ;
1586 */
1587     margin: 0ex .5em 0ex .5em ;
1588     padding: 0 ;
1589 }
1590
1591
1592 figure {
1593     margin: 3ex auto 3ex auto ;
1594     padding: 1ex 1em 1ex 1em ;
1595     overflow-x: auto ;
1596 }
1597
1598
1599 /* To automatically center images in figures: */
1600 /*
1601 figure img.inlineimage {
1602     margin: 0ex auto 0ex auto ;
1603     display: block ;
1604 }
1605 */
1606
```

```
1607 /* To automatically center minipages in figures: */
1608 /*
1609 figure div.minipage, figure div.minipage div.minipage {
1610     margin: 1ex auto 1ex auto ;
1611     display: block ;
1612 }
1613 */
1614
1615 figure div.minipage p { font-size: 85% ; }
1616
1617 figure.subfigure, figure.subtable {
1618     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1619 }
1620
1621 figcaption .minipage { margin:0 ; padding: 0 }
1622
1623 div.minipage figure { border: none ; box-shadow: none ; }
1624
1625 div.floatrow { text-align: center; }
1626
1627 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1628
1629 div.floatfoot { font-size: .85em ;
1630     border-top: 1px solid silver ; line-height: 1.2 ; }
1631
1632 figcaption , .lstlistingtitle {
1633     font-size: .85em ;
1634     text-align: center ;
1635     font-weight: bold ;
1636     margin-top: 1ex ;
1637     margin-bottom: 1ex ;
1638 }
1639
1640 figure.subfigure figcaption, figure.subtable figcaption {
1641     border-bottom: none ; background: none ;
1642 }
1643
1644 div.nonfloatcaption {
1645     margin: 1ex auto 1ex auto ;
1646     font-size: .85em ;
1647     text-align: center ;
1648     font-weight: bold ;
1649 }
1650
1651 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1652 figure div.floatrow div.minipage figcaption {
1653     border: none ;
1654     background: none ;
1655 }
1656
```

```
1657
1658 table {
1659     margin: 1ex auto 1ex auto ;
1660     border-collapse: separate ;
1661     border-spacing: 0px ;
1662     line-height: 1.3 ;
1663 }
1664
1665 tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
1666     margin-bottom: 0ex ; } /* for \hline */
1667
1668 tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
1669     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1670
1671 td {padding: 1ex .5em 1ex .5em ;}
1672
1673 table td.tdl { text-align: left ; vertical-align: middle ; }
1674 table td.tdc { text-align: center ; vertical-align: middle ; }
1675 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1676 table td.tdbang { text-align: center ; vertical-align: middle ; }
1677 table td.tdr { text-align: right ; vertical-align: middle ; }
1678 table td.tdp { text-align: left ; vertical-align: bottom ; }
1679 table td.tdm { text-align: left ; vertical-align: middle ; }
1680 table td.tdb { text-align: left ; vertical-align: top ; }
1681 table td.tdP { text-align: center ; vertical-align: bottom ; }
1682 table td.tdM { text-align: center ; vertical-align: middle ; }
1683 table td.tdB { text-align: center ; vertical-align: top ; }
1684
1685 table td.tvertbarl { border-left: 1px solid black }
1686 table td.tvertbarr { border-right: 1px solid black }
1687
1688
1689 /* for cmidrules: */
1690 table td.tdrule {
1691     border-top: 1px solid #A0A0A0 ;
1692 }
1693
1694 table td.tdrulel {
1695     border-top-left-radius:.5em ;
1696     border-top: 1px solid #A0A0A0 ;
1697 }
1698
1699 table td.tdruler {
1700     border-top-right-radius:.5em ;
1701     border-top: 1px solid #A0A0A0 ;
1702 }
1703
1704 table td.tdrulelr {
1705     border-top-left-radius:.5em ;
1706     border-top-right-radius:.5em ;
```

```
1707     border-top: 1px solid #A0A0A0 ;
1708 }
1709
1710
1711 /* Margins of paragraphs inside table cells: */
1712 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1713     padding-bottom: 1ex ; margin: 0ex ; }
1714 td.tdm p , td.tmbrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1715     padding-bottom: 1ex ; margin: 0ex ; }
1716 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1717     padding-bottom: 1ex ; margin: 0ex ; }
1718
1719 td.tdp , td.tdprule , td.tdP , td.tdPrule
1720     { padding: 0ex .5em 0ex .5em ; }
1721 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1722     { padding: 0ex .5em 0ex .5em ; }
1723 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1724     { padding: 0ex .5em 0ex .5em ; }
1725
1726
1727 /* table notes: */
1728 .tnotes {
1729     margin: 0ex 5% 1ex 5% ;
1730     padding: 0.5ex 1em 0.5ex 1em;
1731     font-size:.85em;
1732     text-align: left ;
1733 }
1734
1735 .tnotes dl dt p {margin-bottom:0px;}
1736
1737 .tnoteitemheader {margin-right: 1em;}
1738
1739
1740 /* for bigdelim */
1741 .ldelim, .rdelim { font-size: 200% }
1742
1743
1744 /* center, flushleft, flushright environments */
1745 div.center{text-align:center;}
1746 div.center table {margin-left:auto;margin-right:auto;}
1747 div.flushleft{text-align:left;}
1748 div.flushleft table {margin-left:0em ; margin-right:auto;}
1749 div.flushright{text-align:right;}
1750 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1751
1752
1753 /* Fancybox */
1754 div.Btrivlist table tr td { padding: .2ex 0em ; }
1755
1756
```

```
1757 /* program listing callouts: */
1758 span.callout {
1759     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1760         Geneva, Verdana, sans-serif ;
1761     border-radius: .5em;
1762     background-color:black;
1763     color:white;
1764     padding:0px .25em 0px .25em;
1765     margin: 0 ;
1766     font-weight: bold;
1767     font-size:.72em ;
1768 }
1769
1770 div.programlisting pre.verbatim span.callout{
1771     font-size: .85em ;
1772 }
1773
1774
1775
1776
1777
1778 div.published
1779 {
1780     text-align: center ;
1781     font-variant: normal ;
1782     font-style: italic ;
1783     font-size: 1em ;
1784     margin: 3ex 0em 3ex 0em ;
1785 }
1786
1787 div.subtitle
1788 {
1789     text-align: center ;
1790     font-variant: normal ;
1791     font-style: italic ;
1792     font-size: 1.25em ;
1793     margin: 3ex 0em 3ex 0em ;
1794 }
1795
1796 div.subtitle p { margin: 1ex ; }
1797
1798 div.author
1799 {
1800     font-variant: normal ;
1801     font-style: normal ;
1802     font-size: 1em ;
1803     margin: 3ex 0em 3ex 0em ;
1804 }
1805
1806 div.oneauthor {
```

```
1807     display: inline-block ;
1808     margin: 3ex 1em 0ex 1em ;
1809 }
1810
1811 /*
1812 div.author table {
1813     margin: 3ex auto 0ex auto ;
1814     background: none ;
1815 }
1816
1817 div.author table tbody tr td { padding: .25ex ; }
1818 */
1819
1820 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1821
1822 div.titledate {
1823     text-align: center ;
1824     font-size: .85em ;
1825     font-style: italic;
1826     margin: 6ex 0em 6ex 0em ;
1827 }
1828
1829
1830 nav.topnavigation{
1831     text-align: left ;
1832     padding: 0.5ex 1em 0.5ex 1em ;
1833 /*     margin: 2ex 0em 3ex 0em ; */
1834     margin: 0 ;
1835     border-bottom: 1px solid silver ;
1836     border-top: 1px solid silver ;
1837     clear:right ;
1838 }
1839
1840 nav.botnavigation{
1841     text-align: left ;
1842     padding: 0.5ex 1em 0.5ex 1em ;
1843 /*     margin: 3ex 0em 2ex 0em ; */
1844     margin: 0 ;
1845     border-top: 1px solid silver ;
1846     border-bottom: 1px solid silver ;
1847     clear:right ;
1848 }
1849
1850
1851 header{
1852     line-height: 1.2 ;
1853     font-size: 1em ;
1854 /*     border-bottom: 2px solid silver ; */
1855     margin: 0px ;
1856     padding: 0ex 1em 0ex 1em ;
```

```
1857     text-align:center ;
1858 }
1859
1860 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1861
1862
1863 footer{
1864     font-size: .85em ;
1865     line-height: 1.2 ;
1866     margin-top: 1ex ;
1867     border-top: 2px solid silver ;
1868     padding: 2ex 1em 2ex 1em ;
1869     clear:right ;
1870     text-align:left ;
1871 }
1872
1873
1874 a.linkhome { font-weight:bold ; font-size: 1em ;}
1875
1876
1877 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
1878
1879 img.lateximage{
1880     padding: 0px 0px 0px 0px ;
1881     box-shadow: none ;
1882     border: none ;
1883     background: none ;
1884     margin: 0px 0px -.15ex 0px ;
1885     /* pdfcrop leaves a slight margin, adjust to baseline */
1886     max-width: 100% ;
1887     border-radius: 0ex ;
1888     border: none ;
1889 }
1890
1891
1892
1893 nav.sidetoc {
1894     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1895         "Lucida Bright", Georgia, serif;
1896     float:right ;
1897     width: 20%;
1898     border-left: 1px solid silver;
1899     border-top: 1px solid silver;
1900     border-bottom: 1px solid silver;
1901 /*     border-top: 2px solid #808080 ; */
1902     background: #FAF7F4 ;
1903     padding: 2ex 0em 2ex 1em ;
1904     margin: 0ex 0em 2ex 1em ;
1905     font-size:.9em ;
1906     border-radius: 20px 0px 0px 20px ;
```

```
1907     }
1908
1909 div.sidetoccontents {
1910 /*     border-top: 1px solid silver ; */
1911     overflow-y: auto ;
1912     width: 100% ;
1913     text-align: left ;
1914 }
1915
1916
1917 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
1918     text-indent: 0 ; }
1919
1920 nav.sidetoc p a {color:black ; font-size: .7em ;}
1921
1922 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
1923     border-bottom: 1px solid silver ;     }
1924
1925 nav.sidetoc a:hover {text-decoration: underline ; }
1926
1927
1928
1929 section.textbody { margin: 0ex 1em 0ex 1em ;}
1930
1931
1932 div.multicolsheading { -webkit-column-span: all;
1933     -moz-column-span: all; column-span: all; }
1934 div.multicols { -webkit-columns: 3 380px ;
1935     -moz-columns: 3 380px ; columns: 3 380px ; }
1936 div.multicols p {margin-top: 0ex}
1937
1938
1939
1940 /* Used to support algorithmicx: */
1941 span.floatright { float: right ; }
1942
1943
1944
1945
1946 /* Native LaTeX theorems: */
1947
1948 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
1949 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
1950
1951
1952 /* theorem, amsthm, and ntheorem packages */
1953
1954 span.theoremheader,
1955 span.theoremheaderplain,
1956 span.theoremheaderdefinition,
```

```
1957 span.theoremheaderbreak,
1958 span.theoremheadermarginbreak,
1959 span.theoremheaderchangebreak,
1960 span.theoremheaderchange,
1961 span.theoremheadermargin
1962 {
1963     font-style:normal ; font-weight: bold ; margin-right: 1em ;
1964 }
1965
1966 span.amsthmnameplain,
1967 span.amsthmnamedefinition,
1968 span.amsthmnumberplain,
1969 span.amsthmnumberdefinition
1970 {
1971     font-style:normal ; font-weight: bold ;
1972 }
1973
1974
1975 span.amsthmnameremark,
1976 span.amsthmnumberremark
1977 {font-style:italic ; font-weight: normal ; }
1978
1979
1980 span.amsthmnoteplain,
1981 span.amsthmnotedefinition
1982 {font-style:normal ;}
1983
1984
1985 span.theoremheaderremark,
1986 span.theoremheaderproof,
1987 span.amsthmproofname
1988 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1989
1990 span.theoremheadersc
1991 {
1992     font-style:normal ;
1993     font-variant: small-caps ;
1994     font-weight: normal ;
1995     margin-right: 1em ;
1996 }
1997
1998 .theoremendmark {float:right}
1999
2000 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonnumberplain,
2001 div.theorembodybreak, div.theorembodynonnumberbreak,
2002 div.theorembodymarginbreak,
2003 div.theorembodychangebreak,
2004 div.theorembodychange,
2005 div.theorembodymargin
2006 {
```

```
2007     font-style:italic;
2008     margin-top: 3ex ; margin-bottom: 3ex ;
2009 }
2010
2011 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
2012 div.theorembodyplainupright, nonumberplainuprightsc,
2013 div.amsthmbodydefinition, div.amsthmbodyremark,
2014 div.amsthmproof
2015 {
2016     font-style: normal ;
2017     margin-top: 3ex ; margin-bottom: 3ex ;
2018 }
2019
2020 span.amsthmnoteremark {}
2021
2022
2023
2024 /*
2025 For CSS LaTeX and related logos:
2026 Based on:
2027 http://edward.oconnor.cx/2007/08/tex-poshlet
2028 http://nitens.org/taraborelli/texlogo
2029 */
2030
2031 .latexlogofont {
2032     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2033         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2034     font-variant: normal ;
2035 }
2036
2037 .latexlogo {
2038     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2039         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2040     letter-spacing: .03em ;
2041     font-size: 1.1em;
2042 }
2043
2044 .latexlogo sup {
2045     text-transform: uppercase;
2046     letter-spacing: .03em ;
2047     font-size: 0.85em;
2048     vertical-align: 0.15em;
2049     margin-left: -0.36em;
2050     margin-right: -0.15em;
2051 }
2052
2053 .latexlogo sub {
2054     text-transform: uppercase;
2055     vertical-align: -0.5ex;
2056     margin-left: -0.1667em;
```

```
2057 margin-right: -0.125em;
2058 font-size: 1em;
2059 }
2060
2061 .xetexlogo {
2062     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2063         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2064     letter-spacing: .03em ;
2065     font-size: 1.1em;
2066 }
2067
2068 /* A smaller gap between Xe and Tex v.s. LaTeX: */
2069 .xetexlogo sub {
2070     text-transform: uppercase;
2071     vertical-align: -0.5ex;
2072     margin-left: -0.0667em;
2073     margin-right: -0.2em;
2074     font-size: 1em;
2075     letter-spacing: .03em ;
2076 }
2077
2078 /* A large gap between Xe and LaTeX v.s. TeX: */
2079 .xelatexlogo sub {
2080     text-transform: uppercase;
2081     vertical-align: -0.5ex;
2082     margin-left: -0.0667em;
2083     margin-right: -.05em;
2084     font-size: 1em;
2085     letter-spacing: .03em ;
2086 }
2087
2088 .amslogo {
2089     font-family: "TeXGyreChorus", "URW Chancery L",
2090         "Apple Chancery", "ITC Zapf Chancery", "Monotype Corsiva",
2091         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
2092         "Hoefler Text", Times, "Times New Roman", serif;
2093     font-style: italic;
2094 }
2095
2096 .lyxlogo {
2097     font-family: "URW Classico", Optima, "Linux Biolinum O",
2098         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2099         Verdana, sans-serif ;
2100 }
2101
2102
2103
2104
2105 /* Only display top and bottom navigation if a small screen: */
2106 /* Hide the sidetoc if a small screen: */
```

```
2107 nav.topnavigation { display:none; }
2108 nav.botnavigation { display:none; }
2109
2110 @media screen and (max-width: 45em) {
2111 /*     nav.sidetoc {display:none;} */
2112     nav.sidetoc {
2113         float: none ;
2114         width: 100% ;
2115         margin: 5ex 0px 5ex 0px ;
2116         padding: 0 ;
2117         border-radius: 0 ;
2118         border-bottom: 1px solid black ;
2119         border-top: 1px solid black ;
2120         box-shadow: none ;
2121     }
2122 /*     nav.topnavigation { display:block } */
2123     nav.botnavigation { display:block }
2124     .marginpar {
2125         max-width: 100%;
2126         float: none;
2127         display:block ;
2128         margin: 1ex 1em 1ex 1em ;
2129     }
2130 }
2131
2132 @media print {
2133     body {
2134         font-family: "Linux Libertine O",
2135             "DejaVu Serif", "Bitstream Vera Serif",
2136             "Liberation Serif", "Nimbus Roman No 9 L",
2137             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2138     }
2139     nav.sidetoc { display:none; }
2140     nav.topnavigation { display: none; }
2141     nav.botnavigation { display: none; }
2142 }
2143
2144 @media handheld {
2145     nav.sidetoc { display:none; }
2146     nav.topnavigation { display:block }
2147     nav.botnavigation { display:block }
2148 }
2149
2150 @media projection {
2151     nav.sidetoc { display:none; }
2152     nav.topnavigation { display:block }
2153     nav.botnavigation { display:block }
2154 }
2155 \end{filecontents*}
2156 % \end{Verbatim}% for syntax highlighting
```

```
2157 \end{warpprint}
```

33.5 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.

```
2158 \begin{warpprint}
2159 \begin{filecontents*}{lwarp_sagebrush.css}
2160 @import url("lwarp.css") ;
2161
2162
2163 A:link {color:#105030 ; text-decoration: none ; }
2164 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2165 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2166 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2167
2168
2169
2170 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2171 {
2172     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2173     "Linux Libertine 0", "Liberation Serif",
2174     "Nimbus Roman No 9 L", "FreeSerif",
2175     "Hoefler Text", Times, "Times New Roman", serif;
2176     font-variant: small-caps ;
2177     font-weight: normal ;
2178     color: #304070 ;
2179     text-shadow: 2px 2px 3px #808080;
2180 }
2181
2182 h1 { /* title of the entire website, used on each page */
2183     font-variant: small-caps ;
2184     color: #304070 ;
2185     text-shadow: 2px 2px 3px #808080;
2186     background-color: #F7F7F0 ;
2187     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2188 }
2189
2190 h1 {
2191     border-bottom: 1px solid #304070;
2192     border-top: 2px solid #304070;
2193 }
2194
2195 h2 {
```

```
2196 border-bottom: 1px solid #304070;
2197 border-top: 2px solid #304070;
2198 background-color: #F7F7F0 ;
2199 background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2200 }
2201
2202
2203
2204 div.abstract {
2205     background: #f5f5eb ;
2206     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2207
2208     border: 1px solid silver;
2209     border-radius: 1em ;
2210 }
2211
2212 div.abstract dl {line-height:1.5;}
2213 div.abstract dt {color:#304070;}
2214
2215 div.abstracttitle{
2216     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2217         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2218         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2219     font-weight:bold;
2220     font-variant: small-caps ;
2221     font-size:1.5em;
2222     border-bottom: 1px solid silver ;
2223     color: #304070 ;
2224     text-align: center ;
2225     text-shadow: 1px 1px 2px #808080;
2226 }
2227
2228 span.abstracrunintitle{
2229     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2230         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2231         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2232     font-weight:bold;
2233 }
2234
2235
2236 div.epigraph, div.dictum {
2237     background: #f5f5eb ;
2238     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2239
2240     border: 1px solid silver ;
2241     border-radius: 1ex ;
2242     box-shadow: 3px 3px 3px #808080 ;
2243 }
2244
2245
```

```
2246 .example {
2247     background-color: #f5f5eb ;
2248     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2249
2250 }
2251
2252 div.exampletitle{
2253     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2254         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2255         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2256     font-weight:bold;
2257     font-variant: small-caps ;
2258     border-bottom: 1px solid silver ;
2259     color: #304070 ;
2260     text-align: center ;
2261     text-shadow: 1px 1px 2px #808080;
2262 }
2263
2264
2265 .sidebar {
2266     background-color: #f5f5eb ;
2267     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2268
2269 }
2270
2271 div.sidebartitle{
2272     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2273         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2274         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2275     font-weight:bold;
2276     font-variant: small-caps ;
2277     border-bottom: 1px solid silver ;
2278     color: #304070 ;
2279     text-align: center ;
2280     text-shadow: 1px 1px 2px #808080;
2281 }
2282
2283
2284 .fancyvrblabel {
2285     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2286         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2287         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2288     font-weight:bold;
2289     font-variant: small-caps ;
2290     font-size: 1.5em ;
2291     color: #304070 ;
2292     text-align: center ;
2293     text-shadow: 1px 1px 2px #808080;
2294 }
2295
```

```
2296
2297
2298 div.minipage {
2299     background-color: #eeeeee7 ;
2300     border: 1px solid silver ;
2301     border-radius: 1ex ;
2302 }
2303
2304 section.textbody > div.minipage {
2305     box-shadow: 3px 3px 3px #808080 ;
2306 }
2307
2308 div.fboxBlock div.minipage { box-shadow: none ; }
2309
2310 .framed .minipage , .framedleftbar .minipage {
2311     border: none ;
2312     background: none ;
2313     padding: 0ex ;
2314     margin: 0ex ;
2315 }
2316
2317 figure.figure .minipage, figcaption .minipage { border: none; }
2318
2319 div.marginblock div.minipage ,
2320 div.marginparblock div.minipage
2321     { border: none; }
2322
2323 figure , div.marginblock {
2324     background-color: #eeeeee7 ;
2325     border: 1px solid silver ;
2326     border-radius: 1ex ;
2327     box-shadow: 3px 3px 3px #808080 ;
2328 }
2329
2330 figure figure {
2331     border: 1px solid silver ;
2332     margin: 0em ;
2333     box-shadow: none ;
2334 }
2335
2336 /*
2337 figcaption {
2338     border-top: 1px solid silver ;
2339     border-bottom: 1px solid silver ;
2340     background-color: #e8e8e8 ;
2341 }
2342 */
2343
2344
2345 div.table {
```

```
2346     box-shadow: 3px 3px 3px #808080 ;
2347 }
2348
2349 /*
2350 .tnotes {
2351     background: #e8e8e8;
2352     border: 1px solid silver;
2353 }
2354 */
2355
2356
2357 nav.topnavigation{
2358     background-color: #b0b8b0 ;
2359     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2360 }
2361
2362 nav.botnavigation{
2363     background-color: #b0b8b0 ;
2364     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2365 }
2366
2367
2368
2369 header{
2370     background-color: #F7F7F0 ;
2371     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2372 }
2373
2374 footer{
2375     background-color: #F7F7F0 ;
2376     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2377 }
2378
2379
2380
2381 nav.sidetoc {
2382     background-color: #F7F7F0 ;
2383     background-image: linear-gradient(to bottom, #F7F7F0, #COCOCO);
2384     box-shadow: 3px 3px 3px #808080 ;
2385     border-radius: 0px 0px 0px 20px ;
2386     }
2387
2388 div.sidetocitle {color: #304070 ; }
2389
2390 nav.sidetoc a:hover {
2391     color:#006000 ;
2392     text-decoration: none ;
2393     text-shadow:0px 0px 2px #a0a0a0;
2394 }
2395
```

```

2396
2397 @media screen and (max-width: 45em) {
2398     nav.sidetoc { border-radius: 0 ; }
2399 }
2400
2401
2402 \end{filecontents*}
2403 % \end{Verbatim}% for syntax highlighting
2404 \end{warpprint}

```

33.6 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.

```

2405 \begin{warpprint}
2406 \begin{filecontents*}{lwarp_formal.css}
2407 @import url("lwarp.css") ;
2408
2409
2410
2411 A:link {color:#802020 ; text-decoration:none; }
2412 A:visited {color:#802020 ; text-shadow:none ;}
2413 A:hover {color:#400000 ; text-shadow:none ;}
2414 A:active {color:#C00000 ; text-shadow:none ;}
2415
2416
2417 body {
2418     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2419         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2420         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2421         "Times New Roman", serif;
2422     background: #fffcf5;
2423 }
2424
2425 span.textrm {
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 span.textsf {
2433     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2434         Geneva, Verdana, sans-serif ;

```

```
2435 }
2436
2437
2438
2439 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2440 {
2441     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2442         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2443         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2444         "Times New Roman", serif;
2445     color: #800000 ;
2446     text-shadow: none ;
2447 }
2448
2449 h1, h2 {
2450     background-color: #fffcf5 ;
2451     background-image: none ;
2452     border-bottom: 1px solid #808080;
2453     border-top: 2px solid #808080;
2454 }
2455
2456 div.abstracttitle {
2457     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2458         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2459         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2460         "Times New Roman", serif;
2461     color: black ;
2462     text-shadow: none ;
2463 }
2464
2465 span.abstracruntintitle {
2466     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2467         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2468         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2469         "Times New Roman", serif;
2470     color: black ;
2471     text-shadow: none ;
2472 }
2473
2474 div.abstract { font-size: 100% }
2475
2476 .sidebar {
2477     background: #fffcf5;
2478     background-image: none ;
2479     margin: 2em 5% 2em 5%;
2480     padding: 0.5em 1em;
2481     border: none ;
2482     border-top : 1px solid silver;
2483     border-bottom : 1px solid silver;
2484     font-size: 90% ;
```

```
2485 }
2486
2487 div.sidebar{
2488     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2489         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2490         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2491         "Times New Roman", serif;
2492     color: #800000 ;
2493     text-shadow: none ;
2494     border: none ;
2495 }
2496
2497 .example {
2498     background: #fffcf5;
2499     background-image: none ;
2500     margin: 2em 5% 2em 5%;
2501     padding: 0.5em 1em;
2502     border: none ;
2503     border-top : 1px solid silver;
2504     border-bottom : 1px solid silver;
2505 }
2506
2507 div.exampletitle{
2508     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2509         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2510         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2511         "Times New Roman", serif;
2512     color: #800000 ;
2513     text-shadow: none ;
2514     border: none ;
2515 }
2516
2517 div.fancyvlabel{
2518     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2519         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2520         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2521         "Times New Roman", serif;
2522     color: #800000 ;
2523     text-shadow: none ;
2524     border: none ;
2525 }
2526
2527
2528
2529 .verse {
2530     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2531         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2532         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2533         "Times New Roman", serif;
2534 }
```

```
2535
2536
2537 figure {
2538     margin: 3ex 5% 3ex 5% ;
2539     padding: 1ex 1em 1ex 1em ;
2540     background-color: #fffcf5 ;
2541     overflow-x: auto ;
2542     border: none ;
2543 /*     border-top: 1px solid silver; */
2544 /*     border-bottom: 1px solid silver; */
2545 }
2546
2547
2548 figcaption , .lstlisting {
2549     border: none ;
2550 /*     border-top: 1px solid silver ; */
2551 /*     border-bottom: 1px solid silver ; */
2552     background-color: #fffcf5 ;
2553 }
2554
2555 .tnotes {
2556     background: #fffcf5 ;
2557 }
2558
2559 .theorem {
2560     background: none ;
2561 }
2562
2563 .minipage {
2564     background-color: #fffcf5 ;
2565     border: none ;
2566 }
2567
2568 div.floatrow figure { border: none ; }
2569
2570 figure figure { border: none ; }
2571
2572
2573 nav.toc, nav.lof, nav.lot, nav.lol {
2574     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2575     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2576     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2577     "Times New Roman", serif;
2578 }
2579
2580 nav.sidetoc {
2581     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2582     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2583     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2584     "Times New Roman", serif;
```

```

2585     background-image: linear-gradient(to bottom, #ffcf5, #C0C0C0);
2586     border-radius: 0px 0px 0px 20px ;
2587 }
2588
2589 div.sidetocitle{
2590     color: #800000 ;
2591 }
2592
2593 header{
2594     background-color: #e0e0e0 ;
2595     background-image: linear-gradient(to top, #ffcf5, #b0b0b0);
2596     text-align:center ;
2597 }
2598
2599 footer{
2600     background-color: #e0e0e0 ;
2601     background-image: linear-gradient(to bottom, #ffcf5, #b0b0b0);
2602     padding: 2ex 1em 2ex 1em ;
2603     clear:right ;
2604     text-align:left ;
2605 }
2606
2607 nav.botnavigation {
2608     background: #dedcd5 ;
2609     border-top: 1px solid black ;
2610 }
2611 \end{filecontents*}
2612 % \end{Verbatim}% for syntax highlighting
2613 \end{warpprint}

```

33.7 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.

```

2614 \begin{warpprint}
2615 \begin{filecontents*}{sample_project.css}
2616 /* ( --- Start of project.css --- ) */
2617 /* ( --- A sample project-specific CSS file for lwarp --- ) */
2618
2619 /* Load default lwarp settings: */
2620 @import url("lwarp.css") ;
2621 /* or lwarp_formal.css, lwarp_sagebrush.css */
2622
2623 /* Project-specific CSS setting follow here. */

```

```

2624 /* . . . */
2625
2626 /* ( --- End of project.css --- ) */
2627 \end{filecontents*}
2628 % \end{Verbatim}% for syntax highlighting
2629 \end{warpprint}

```

33.8 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.

```

2630 \begin{warpprint}
2631 \begin{filecontents*}{lwarp.xdy}
2632 (require "tex/inputenc/latin.xdy")
2633 (merge-rule "\\PS *" "Postscript")
2634 (require "texindy.xdy")
2635 (require "page-ranges.xdy")
2636 (require "book-order.xdy")
2637 (markup-locref :open "\hyperindexref{" :close "}")
2638 \end{filecontents*}
2639 % \end{Verbatim}% for syntax highlighting
2640 \end{warpprint}

```

33.9 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.

```

2641 \begin{warpprint}
2642 \begin{filecontents*}{lwarp_mathjax.txt}
2643 <!-- https://groups.google.com/forum/#!topic/
2644         mathjax-users/jUtewUcE2bY -->
2645 <script type="text/x-mathjax-config">
2646 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2647     var seteqsectionDefault = {name: "", num: 0};
2648     var seteqsections = {}, seteqsection = seteqsectionDefault;
2649     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2650     var AMS = MathJax.Extension["TeX/AMSmath"];
2651     TEX.Definitions.Add({
2652     macros: {

```

```
2653     seteqsection: "mySection",
2654     seteqnumber: "mySetEqNumber"
2655   }
2656   });
2657
2658   PARSE.Augment({
2659   mySection: function (name) {
2660     seteqsection.num = AMS.number;
2661     var n = this.GetArgument(name);
2662     if (n === "") {
2663       seteqsection = seteqsectionDefault;
2664     } else {
2665       if (!seteqsections["_"+n])
2666         seteqsections["_"+n] = {name:n, num:0};
2667       seteqsection = seteqsections["_"+n];
2668     }
2669     AMS.number = seteqsection.num;
2670   },
2671   mySetEqNumber: function (name) {
2672     var n = this.GetArgument(name);
2673     if (!n || !n.match(/^ *[0-9]+ *$/))
2674       n = ""; else n = parseInt(n)-1;
2675     <!-- $ syntax highlighting -->
2676     if (n === "" || n < 1)
2677       TEX.Error
2678         ("Argument to "+name+" should be a positive integer");
2679     AMS.number = n;
2680   }
2681   });
2682   MathJax.Hub.Config({
2683   TeX: {
2684     equationNumbers: {
2685       formatTag: function (n)
2686         {return "("+(seteqsection.name+"."+n).replace(/\./, "")+"}"},
2687       formatID: function (n) {
2688         n = (seteqsection.name+'.'+n).replace
2689           (/[:'>&]/g, "").replace(/\./, "");
2690         return 'mjx-eqn-' + n;
2691       }
2692     }
2693   }
2694   });
2695 });
2696 </script>
2697
2698 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2699 <script type="text/x-mathjax-config">
2700   MathJax.Ajax.config.path["Contrib"] =
2701     "https://cdn.mathjax.org/mathjax/contrib";
2702 </script>
```

```

2703
2704 <!-- https://github.com/burnpanck/MathJax-siunitx -->
2705
2706 <script type="text/x-mathjax-config">
2707   MathJax.Hub.Config({
2708     extensions: ["tex2jax.js","[siunitx]/siunitx.js"],
2709     jax: ["input/TeX","output/HTML-CSS"],
2710     tex2jax: {inlineMath: [["$","$"],["\\(", "\\)"]]},
2711     TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "siunitx.js"]}
2712   });
2713   MathJax.Ajax.config.path['siunitx'] = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
2714 </script>
2715
2716 <script type="text/x-mathjax-config">
2717   MathJax.Hub.Config({
2718     TeX: {
2719       equationNumbers: {
2720         autoNumber: "AMS"
2721       }
2722     }
2723   });
2724 </script>
2725
2726 <!-- Alternative CDN provider: -->
2727 <script type="text/javascript" async
2728   src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2729 </script>
2730
2731 <!-- No longer supported after April 30, 2017: -->
2732 <!--
2733 <script
2734   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2735 </script>
2736 -->
2737
2738 \end{filecontents*}
2739 % \end{Verbatim}% for syntax highlighting
2740 \end{warpprint}

```

33.10 lwarpmk option

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

```
2741 \begin{LWR@createlwarpmk}
```

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

```
2742 \begin{filecontents*}{lwarpmk.lua}
2743 #!/usr/bin/env texlua
2744
2745 -- Copyright 2016-2017 Brian Dunn
2746
2747 -- Print the usage of the lwarpmk command:
2748
2749 printversion = "v0.44"
2750
2751 function printhelp ()
2752 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2753 end
2754
2755 function printusage ()
2756 print ( [[
2757
2758 lwarpmk print [project]: Compile the print version if necessary.
2759 lwarpmk print1 [project]: Forced single compile of the print version.
2760 lwarpmk printindex [project]: Process the index for the print version.
2761 lwarpmk printglossary [project]: Process the glossary for the print version.
2762 lwarpmk html [project]: Compile the HTML version if necessary.
2763 lwarpmk html1 [project]: Forced single compile of the HTML version.
2764 lwarpmk htmlindex [project]: Process the index for the html version.
2765 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2766 lwarpmk again [project]: Touch the source code to trigger recompiles.
2767 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2768 lwarpmk pdftohtml [project]:
2769     For use with latexmk or a Makefile:
2770     Convert project_html.pdf to project_html.html and
2771     individual HTML files.
2772 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
2773 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2774 lwarpmk -h: Print this help message.
2775 lwarpmk --help: Print this help message.
2776
2777 ]] )
2778 printconf ()
2779 end
2780
2781 -- Print the format of the configuration file lwarpmk.conf:
2782
2783 function printconf ()
2784 print ( [[
2785 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2786 --
2787 opsystem = "Unix"    (or "Windows")
2788 latexname = "pdflatex" (or "lualatex", or "xelatex")
2789 sourcename = "projectname" (the source-code filename w/o .tex)
2790 homehtmlfilename = "index" (or perhaps the project name)
2791 htmlfilename = "" (or "projectname" - filename prefix)
```

```
2792 latexmk = "false" (or "true" to use latexmk to build PDFs)
2793 language = "english" (use a language supported by xindy)
2794 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2795 --
2796 Filenames must contain only letters, numbers, underscore, or dash.
2797 Values must be in "quotes".
2798
2799 ]] ) ;
2800 end
2801
2802
2803 -- Split one large sourcefile into a number of files,
2804 -- starting with destfile.
2805 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2806
2807 function splitfile (destfile,sourcefile)
2808 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2809 local sfile = io.open(sourcefile)
2810 io.output(destfile)
2811 for line in sfile:lines() do
2812 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|") ;
2813 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2814 io.output(newfilename) ;
2815 else -- not a splitpoint
2816 io.write (line .. "\n") ;
2817 end
2818 end -- do
2819 io.close(sfile)
2820 end -- function
2821
2822 -- Incorrect value, so print an error and exit.
2823
2824 function cvalueerror ( line, linenum , cvalue )
2825     print ( linenum .. " : " .. line ) ;
2826     print ("lwarpmk: incorrect variable value \" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2827     printconf () ;
2828     os.exit(1) ;
2829 end
2830
2831 -- Load settings from the project's "lwarpmk.conf" file:
2832
2833 function loadconf ()
2834 -- Default configuration filename:
2835 local conffile = "lwarpmk.conf"
2836 -- Optional configuration filename:
2837 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2838 -- Default language:
2839 language = "english"
2840 -- Default xdyfile:
2841 xdyfile = "lwarp.xdy"
```

```
2842 -- Verify the file exists:
2843 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2844 print("lwarpmk: " .. conffile .. " does not exist.")
2845 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2846 printhelp () ;
2847 os.exit(1) -- exit the entire lwarpmk script
2848 else -- file exists
2849 -- Read the file:
2850 print ("lwarpmk: Reading " .. conffile .. ".")
2851 local cfile = io.open(conffile)
2852 -- Scan each line:
2853 local linenum = 0
2854 for line in cfile:lines() do -- scan lines
2855 linenum = linenum + 1
2856 i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*"([%w%-_.]*)\("\")" ) ;
2857 -- Error if incorrect enclosing characters:
2858 if ( i == nil ) then
2859 print ( linenum .. " : " .. line ) ;
2860 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2861 printconf () ;
2862 os.exit(1) ;
2863 end
2864 if ( cvarname == "opssystem" ) then
2865 -- Verify choice of opssystem:
2866 if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2867 opssystem = cvalue
2868 else
2869 cvalueerror ( line, linenum , cvalue )
2870 end
2871 elseif ( cvarname == "latexname" ) then
2872 -- Verify choice of LaTeX compiler:
2873 if (
2874 (cvalue == "pdflatex") or
2875 (cvalue == "xelatex") or
2876 (cvalue == "lualatex")
2877 ) then
2878 latexname = cvalue
2879 else
2880 cvalueerror ( line, linenum , cvalue )
2881 end
2882 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2883 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2884 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2885 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2886 elseif ( cvarname == "language" ) then language = cvalue
2887 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2888 else
2889 print ( linenum .. " : " .. line ) ;
2890 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
2891 printconf () ;
```

```
2892 os.exit(1) ;
2893 end
2894 end -- do scan lines
2895 io.close(cfile)
2896 end -- file exists
2897 -- Select some operating-system commands:
2898 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2899   rmname = "rm"
2900   mvname = "mv"
2901   touchnamepre = "touch"
2902   touchnamepost = ""
2903   dirslash = "/"
2904   opquote= "\'"
2905 elseif opsystem=="Windows" then -- For Windows
2906   rmname = "DEL"
2907   mvname = "MOVE"
2908   touchnamepre = "COPY /b"
2909   touchnamepost = "+,,"
2910   dirslash = "\\"
2911   opquote= "\"
2912 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2913 end --- for Windows
2914
2915 -- set xindycmd according to pdflatex vs xelatex/lualatex:
2916 if ( latexname == "pdflatex" ) then
2917   xindycmd = "texindy -C utf8"
2918   glossarycmd = "xindy -C utf8"
2919 else
2920   xindycmd = "xindy -M texindy -C utf8"
2921   glossarycmd = "xindy -C utf8"
2922 end
2923
2924 end -- loadconf
2925
2926
2927 function refreshdate ()
2928 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2929 end
2930
2931
2932 -- Scan the LaTeX log file for the phrase "Rerun to get",
2933 -- indicating that the file should be compiled again.
2934 -- Return true if found.
2935
2936 function reruntoget (filesorce)
2937 local fsource = io.open(filesorce)
2938 for line in fsource:lines() do
2939   if ( string.find(line,"Rerun to get") ~= nil ) then
2940     io.close(fsource)
2941     return true
```

```
2942 end
2943 end
2944 io.close(fsource)
2945 return false
2946 end
2947
2948
2949 -- Compile one time, return true if should compile again.
2950 -- fsuffix is "" for print, "_html" for HTML output.
2951
2952 function onetime (fsuffix)
2953 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2954 err = os.execute(
2955 --      "echo " ..
2956      latexname .. " " .. sourcename..fsuffix )
2957 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2958 return (reruntoget(sourcename .. fsuffix .. ".log" ) ) ;
2959 end
2960
2961
2962 -- Compile up to five times.
2963 -- fsuffix is "" for print, "_html" for HTML output
2964
2965 function manytimes (fsuffix)
2966 if onetime(fsuffix) == true then
2967 if onetime(fsuffix) == true then
2968 if onetime(fsuffix) == true then
2969 if onetime(fsuffix) == true then
2970 if onetime(fsuffix) == true then
2971 end end end end end
2972 end
2973
2974 -- Exit if the given file does not exist.
2975
2976 function verifyfileexists (filename)
2977 if (lfs.attributes ( filename , "modification" ) == nil ) then
2978 print ( "lwarpmk: " .. filename .. " not found." ) ;
2979 os.exit (1) ;
2980 end
2981 end
2982
2983
2984 -- Convert <project>_html.pdf into HTML files:
2985
2986 function pdftohtml ( )
2987 -- Convert to text:
2988 print ("lwarpmk: Converting " .. sourcename
2989      .. "_html.pdf to " .. sourcename .. "_html.html")
2990 os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2991      .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
```

```

2992 -- Split the result into individual HTML files:
2993 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2994 end
2995
2996
2997 -- Remove auxiliary files:
2998
2999 function removeaux ()
3000     os.execute ( rmname .. " " ..
3001         sourcename .. ".aux " .. sourcename .. "_html.aux " ..
3002         sourcename .. ".toc " .. sourcename .. "_html.toc " ..
3003         sourcename .. ".lof " .. sourcename .. "_html.lof " ..
3004         sourcename .. ".lot " .. sourcename .. "_html.lot " ..
3005         sourcename .. ".idx " .. sourcename .. "_html.idx " ..
3006         sourcename .. ".ind " .. sourcename .. "_html.ind " ..
3007         sourcename .. ".log " .. sourcename .. "_html.log " ..
3008         sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
3009         "*_html_inc.*"
3010     )
3011 end
3012
3013
3014
3015 -- Create lateximages based on lateximages.txt:
3016 function createlateximages ()
3017 print ("lwarpmk: Creating lateximages.")
3018 local limagesfile = io.open("lateximages.txt")
3019 -- Create the lateximages directory, ignore error if already exists
3020 err = os.execute("mkdir lateximages")
3021 -- Scan lateximages.txt
3022 for line in limagesfile:lines() do
3023 -- lwimgpage is the page number in the PDF which has the image
3024 -- lwimgnum is the sequential lateximage number to assign for the image
3025 i,j,lwimgpage,lwimgnum = string.find (line,"|(.)|(.)|")
3026 -- For each entry:
3027 if ( i~=nil ) then
3028 -- Separate out the image into its own single-page pdf:
3029 err = os.execute(
3030 "pdfseparate -f " .. lwimgpage .. " -1 " ..
3031 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
3032 -- Crop the image:
3033 err = os.execute(
3034 "pdfcrop -- hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
3035 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3036 -- Convert the image to svg:
3037 err = os.execute(
3038 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
3039 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3040 -- Move the result into lateximages/:
3041 err = os.execute(

```

```
3042 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
3043 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3044 -- Remove the temporary files:
3045 err = os.execute(
3046 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
3047 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3048 end
3049 end -- do
3050 io.close(imagesfile)
3051 end -- function
3052
3053
3054 -- Use latexmk to compile source and index:
3055 -- fsuffix is "" for print, or "_html" for HTML
3056 function compilelatexmk ( fsuffix )
3057     -- The recorder option is required to detect changes in <project>.tex
3058     -- while we are loading <project>_html.tex.
3059     err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
3060         .. "-e "
3061         .. opquote
3062         .. "$makeindex = q/" -- $
3063         .. xindycmd
3064         .. " -M " .. xdyfile
3065         .. " -L " .. language .. " /"
3066         .. opquote
3067         .. " -pdflatex=\\"" .. latexname .. "%O %S\" "
3068         .. sourcename..fsuffix .. ".tex" ) ;
3069     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
3070 end
3071
3072
3073
3074 -- lwarpmk --version :
3075
3076 if (arg[1] == "--version") then
3077 print ( "lwarpmk: " .. printversion )
3078
3079 else -- not --version
3080
3081 -- print intro:
3082
3083 print ( "lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package." )
3084
3085 -- lwarpmk print:
3086
3087 if arg[1] == "print" then
3088 loadconf ( )
3089 if ( latexmk == "true" ) then
3090     compilelatexmk ( "" )
3091     print ( "lwarpmk: Done." )
```

```
3092 else -- not latexmk
3093   verifyfileexists (sourcename .. ".tex") ;
3094   -- See if up to date:
3095   if (
3096     ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
3097     (
3098       lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3099       lfs.attributes ( sourcename .. ".pdf" , "modification" )
3100     )
3101   ) then
3102     -- Recompile if not yet up to date:
3103     manytimes("")
3104     print ("lwarpmk: Done.") ;
3105   else
3106     print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
3107   end
3108 end -- not latexmk
3109
3110 elseif arg[1] == "print1" then
3111   loadconf ()
3112   verifyfileexists (sourcename .. ".tex") ;
3113   onetime("")
3114   print ("lwarpmk: Done.") ;
3115
3116 -- lwarp printindex:
3117 -- Compile the index then touch the source
3118 -- to trigger a recompile of the document:
3119
3120 elseif arg[1] == "printindex" then
3121 loadconf ()
3122 print ("lwarpmk: Processing the index.")
3123 os.execute(
3124   xindycmd
3125   .. " -M " .. xdyfile
3126   .. " -L " .. language
3127   .. " " .. sourcename .. ".idx")
3128 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3129 refreshdate ()
3130 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3131 print ("lwarpmk: Done.")
3132
3133 -- lwarp printglossary:
3134 -- Compile the glossary then touch the source
3135 -- to trigger a recompile of the document:
3136
3137 elseif arg[1] == "printglossary" then
3138 loadconf ()
3139 print ("lwarpmk: Processing the glossary.")
3140
3141 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
```

```
3142 " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
3143 .. sourcename .. ".glo")
3144 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3145 refreshdate ()
3146 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3147 print ("lwarpmk: Done.")
3148
3149 -- lwarpmk html:
3150
3151 elseif arg[1] == "html" then
3152 loadconf ()
3153 if ( latexmk == "true" ) then
3154   compilelatexmk ("_html")
3155   pdftohtml ()
3156   print ("lwarpmk: Done.")
3157 else -- not latexmk
3158   verifyfileexists ( sourcename .. ".tex" ) ;
3159   -- See if exists and is up to date:
3160   if (
3161     ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3162     (
3163       lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3164       lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3165     )
3166   ) then
3167     -- Recompile if not yet up to date:
3168     manytimes("_html")
3169     pdftohtml ()
3170     print ("lwarpmk: Done.")
3171   else
3172     print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3173   end
3174 end -- not latexmk
3175
3176 elseif arg[1] == "html1" then
3177   loadconf ()
3178   verifyfileexists ( sourcename .. ".tex" ) ;
3179   onetime("_html")
3180   pdftohtml ()
3181   print ("lwarpmk: Done.")
3182
3183 elseif arg[1] == "pdftohtml" then
3184   loadconf ()
3185   pdftohtml ()
3186
3187 -- lwarpmk htmlindex:
3188 -- Compile the index then touch the source
3189 -- to trigger a recompile of the document:
3190
3191 elseif arg[1] == "htmlindex" then
```

```
3192 loadconf ()
3193 print ("lwarpmk: Processing the index.")
3194 os.execute(
3195     xindycmd
3196     .. " -M " .. xdyfile
3197     .. " -L " .. language
3198     .. " " .. sourcename .. "_html.idx"
3199 )
3200 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3201 refreshdate ()
3202 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3203 print ("lwarpmk: Done.")
3204
3205 -- lwarpmk htmlglossary:
3206 -- Compile the glossary then touch the source
3207 -- to trigger a recompile of the document:
3208
3209 elseif arg[1] == "htmlglossary" then
3210 loadconf ()
3211 print ("lwarpmk: Processing the glossary.")
3212
3213 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3214     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3215     "_html.gls " .. sourcename .. "_html.glo")
3216
3217 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3218 refreshdate ()
3219 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3220 print ("lwarpmk: Done.")
3221
3222 -- lwarpmk limages:
3223 -- Scan the lateximages.txt file to create lateximages,
3224 -- then touch the source to trigger a recompile.
3225
3226 elseif arg[1] == "limages" then
3227 loadconf ()
3228 print ("lwarpmk: Processing images.")
3229 createlateximages ()
3230 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3231 refreshdate ()
3232 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3233 print ("lwarpmk: Done.")
3234
3235 -- lwarpmk again:
3236 -- Touch the source to trigger a recompile.
3237
3238 elseif arg[1] == "again" then
3239 loadconf ()
3240 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3241 refreshdate ()
```

```
3242 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3243 print ("lwarpmk: Done.")
3244
3245 -- lwarpmk clean:
3246 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3247
3248 elseif arg[1] == "clean" then
3249 loadconf ()
3250 removeaux ()
3251 print ("lwarpmk: Done.")
3252
3253 -- lwarpmk cleanall
3254 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3255 -- and also project.pdf, *.html
3256
3257 elseif arg[1] == "cleanall" then
3258 loadconf ()
3259 removeaux ()
3260 os.execute ( rmname .. " " ..
3261     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
3262     "*.html"
3263     )
3264 print ("lwarpmk: Done.")
3265
3266 -- lwarpmk with no argument :
3267
3268 elseif (arg[1] == nil) then
3269 printhelp ()
3270
3271 -- lwarpmk -h or lwarpmk --help :
3272
3273 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3274 printusage ()
3275
3276 else
3277 print ("lwarpmk: Unknown command \"" .. arg[1] .. "\".\n")
3278 printhelp ()
3279 end
3280
3281 end -- not --version
3282 \end{filecontents*}
3283 % \end{Verbatimim}% for syntax highlighting

3284 \end{LWR@createlwarpmk}
```

34 Stacks

for HTML output: 3285 `\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.

34.1 Assigning depths

initial depths for empty stack entries:

```
3286 \newcommand*\LWR@depthnone}{-5}
```

all sectioning depths are deeper than `LWR@depthfinished`:

```
3287 \newcommand*\LWR@depthfinished}{-4}
3288 \newcommand*\LWR@depthpart}{-1}
3289 \newcommand*\LWR@depthchapter}{0}
3290 \newcommand*\LWR@depthsection}{1}
3291 \newcommand*\LWR@depthsubsection}{2}
3292 \newcommand*\LWR@depthsubsubsection}{3}
3293 \newcommand*\LWR@depthparagraph}{4}
3294 \newcommand*\LWR@depthsubparagraph}{5}
```

used by `\itemize`, `\enumerate`, `\description`:

```
3295 \newcommand*\LWR@depthlist}{6}
```

used by `\item`:

```
3296 \newcommand*\LWR@depthlistitem}{7}
```

34.2 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.

```

3297 \newcommand*\LWR@closeone}{}% top of the stack
3298 \newcommand*\LWR@closetwo}{
3299 \newcommand*\LWR@closethree}{
3300 \newcommand*\LWR@closefour}{
3301 \newcommand*\LWR@closefive}{
3302 \newcommand*\LWR@closesix}{
3303 \newcommand*\LWR@closeseven}{
3304 \newcommand*\LWR@closeeight}{
3305 \newcommand*\LWR@closenine}{
3306 \newcommand*\LWR@closeten}{
3307 \newcommand*\LWR@closeeleven}{
3308 \newcommand*\LWR@closetwelve}{

```

34.3 Closing depths

A stack to record the depth of each level:



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

Ex:

```

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

```

```

3309 \newcommand*\LWR@closedepthone}{\LWR@depthnone}% top of the stack
3310 \newcommand*\LWR@closedepthtwo}{\LWR@depthnone}
3311 \newcommand*\LWR@closedepththree}{\LWR@depthnone}
3312 \newcommand*\LWR@closedepthfour}{\LWR@depthnone}
3313 \newcommand*\LWR@closedepthfive}{\LWR@depthnone}
3314 \newcommand*\LWR@closedepthsix}{\LWR@depthnone}
3315 \newcommand*\LWR@closedepthseven}{\LWR@depthnone}
3316 \newcommand*\LWR@closedeptheight}{\LWR@depthnone}
3317 \newcommand*\LWR@closedepthnine}{\LWR@depthnone}
3318 \newcommand*\LWR@closedephten}{\LWR@depthnone}

```

```
3319 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3320 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

34.4 Pushing and popping the stack

`\pushclose` $\langle action \rangle$ $\langle depth \rangle$

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

```
3321 \NewDocumentCommand{\pushclose}{m m}
3322 {
3323 \let\LWR@closetwelve\LWR@closeeleven
3324 \let\LWR@closeeleven\LWR@closeten
3325 \let\LWR@closeten\LWR@closenine
3326 \let\LWR@closenine\LWR@closeeight
3327 \let\LWR@closeeight\LWR@closeseven
3328 \let\LWR@closeseven\LWR@closesix
3329 \let\LWR@closesix\LWR@closefive
3330 \let\LWR@closefive\LWR@closefour
3331 \let\LWR@closefour\LWR@closethree
3332 \let\LWR@closethree\LWR@closetwo
3333 \let\LWR@closetwo\LWR@closeone
3334 \let\LWR@closeone#1
3335 \let\LWR@closedephtwelve\LWR@closedeptheleven
3336 \let\LWR@closedepthelevel\LWR@closedephten
3337 \let\LWR@closedephten\LWR@closedepthnine
3338 \let\LWR@closedepthnine\LWR@closedeptheight
3339 \let\LWR@closedeptheight\LWR@closedepthseven
3340 \let\LWR@closedepthseven\LWR@closedepthsix
3341 \let\LWR@closedepthsix\LWR@closedepthfive
3342 \let\LWR@closedepthfive\LWR@closedepthfour
3343 \let\LWR@closedepthfour\LWR@closedepththree
3344 \let\LWR@closedepththree\LWR@closedephtwo
3345 \let\LWR@closedephttwo\LWR@closedepthone
3346 \let\LWR@closedepthone#2
3347 }
```

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

```
3348 \newcommand*{\popclose}
3349 {
3350 \let\LWR@closeone\LWR@closetwo
3351 \let\LWR@closetwo\LWR@closethree
3352 \let\LWR@closethree\LWR@closefour
3353 \let\LWR@closefour\LWR@closefive
3354 \let\LWR@closefive\LWR@closesix
3355 \let\LWR@closesix\LWR@closeseven
```

```

3356 \let\LWR@closeseven\LWR@closeeight
3357 \let\LWR@closeeight\LWR@closenine
3358 \let\LWR@closenine\LWR@closeten
3359 \let\LWR@closeten\LWR@closeeleven
3360 \let\LWR@closeeleven\LWR@closetwelve
3361 \let\LWR@closedepthone\LWR@closedepthtwo
3362 \let\LWR@closedepthtwo\LWR@closedepththree
3363 \let\LWR@closedepththree\LWR@closedepthfour
3364 \let\LWR@closedepthfour\LWR@closedepthfive
3365 \let\LWR@closedepthfive\LWR@closedepthsix
3366 \let\LWR@closedepthsix\LWR@closedepthseven
3367 \let\LWR@closedepthseven\LWR@closedeptheight
3368 \let\LWR@closedeptheight\LWR@closedepthnine
3369 \let\LWR@closedepthnine\LWR@closedephten
3370 \let\LWR@closedephten\LWR@closedeptheleven
3371 \let\LWR@closedeptheleven\LWR@closedepthtwelve
3372 }

3373 \end{warpHTML}

```

35 Data arrays

These macros are similar to the arrayjobx package, except that `\LWR@setexpparray`'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@setexpparray{name}{index}{}
```

for HTML output: 3374 `\begin{warpHTML}`

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

```

3375 \NewDocumentCommand{\LWR@setexpparray}{m m m}{%
3376 \ifstrempy{#3}%
3377 {\csdef{#1#2}{}}%
3378 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3379 }

```

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

```
3380 \newcommand*{\LWR@getexpparray}[2]{\csuse{#1#2}}
```

```
3381 \end{warpHTML}
```

36 HTML entities

for HTML output: 3382 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
3383 \let\LWR@origampersand\&
```

\HTMLentity {<entitytag>}

```
3384 \newcommand*{\HTMLentity}[1]{%
3385 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3386 \begingroup%
3387 \LWR@FBcancel%
3388 \LWR@origampersand#1;%
3389 \endgroup
3390 % \LWR@traceinfo{HTMLentity done}%
3391 }
```

\HTMLunicode {<hex_unicode>}

```
3392 \newcommand*{\HTMLunicode}[1]{\HTMLentity{#x#1}}
```

\&

```
3393 \renewcommand*{\&}{\HTMLentity{amp}}
```

\textless
\textgreater

```
3394 \let\LWR@origtextless\textless
3395 \renewcommand*{\textless}{\HTMLentity{lt}}
3396
3397 \let\LWR@origtextgreater\textgreater
3398 \renewcommand*{\textgreater}{\HTMLentity{gt}}
```

```
3399 \end{warpHTML}
```

37 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: 3400 `\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.

```
3401 \providecommand*\BaseJobname{\jobname}
```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section 7.3.1.

```
3402 \providecommand*\HTMLFilename{}
```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 7.3.1.

```
3403 \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.

```
3404 \newcommand*\SetHTMLFileNumber[1]{%
3405 \setcounter{LWR@htmlfilenumber}{#1}%
3406 }
```

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

Defaults to use section names in the filenames.

```
3407 \newbool{FileSectionNames}
3408 \booltrue{FileSectionNames}
```

```
3409 \end{warpall}
```

for HTML output: 3410 `\begin{warpHTML}`

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

```
3411 \newcounter{LWR@htmlfilenumber}
3412 \setcounter{LWR@htmlfilenumber}{0}
```

\LWR@htmlsectionfilename *{(htmlfilenumber or name)}*

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

```
3413 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3414 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
3415 \LWR@traceinfo{about to assign temp}%
3416 \edef\LWR@tempone{#1}%
3417 \LWR@traceinfo{about to compare with ??}%
3418 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3419 {\LWR@traceinfo{found ??}}%
3420 {\LWR@traceinfo{not found ??}}%
3421 \LWR@traceinfo{about to compare with zero or empty}%
3422 \ifthenelse{%
3423   \equal{\LWR@tempone}{0}%
3424   \OR \equal{\LWR@tempone}{ }%
3425   \OR \equal{\LWR@tempone}{??}}%
3426 }%
3427 {%
3428   \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3429   \HomeHTMLFilename.html%
3430 }%
```

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

```
3431 {%
3432   \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
3433   \ifthenelse{%
3434     \equal{\HTMLFilename}{ } \AND
3435     \equal{\LWR@tempone}{Index} \OR
3436     \equal{\LWR@tempone}{index}}%
3437   }%
3438   {%
3439     \LWR@traceinfo{Prefixing the index name with an underscore.}%
3440     \_#1.html%
3441   }%
```

Otherwise, create a filename with the chosen prefix:

```
3442   {\HTMLFilename#1.html}%
3443 }%
3444 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
3445 }
```

`\LWR@htmlrefsectionfilename` `{\label}`

Prints the filename for the given label

```
3446 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3447 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
3448 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
3449 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
3450 }

3451 \end{warpHTML}
```

38 Homepage link

for HTML output: 3452 `\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.

```
3453 \newcommand*{\LinkHome}{%
3454 \LWR@subhyperrefclass{%
3455 \HomeHTMLFilename.html}%
3456 {Home}{linkhome}%
3457 }
```

`\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.

```
3458 \newcommand*{\LWR@topnavigation}{
3459 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
3460 }
```

`\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.

```
3461 \newcommand*{\LWR@botnavigation}{
```

```

3462 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
3463 }

3464 \end{warpHTML}

```

39 \PrintStack diagnostic tool



Diagnostics tool: Prints the \TeX 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: 3465 `\begin{warpHTML}`

`\PrintStack` Prints the closedepth stack.

```

3466 \newcommand*{\PrintStack}{
3467 \LWR@startpars
3468 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3469 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3470 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3471 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
3472 }

3473 \end{warpHTML}

```

40 Closing stack levels

for HTML output: 3474 `\begin{warpHTML}`

Close one nested level:

```

3475 \newcommand*{\LWR@closeoneprevious}{%
3476
3477 \LWR@closeone
3478
3479 \popclose
3480 }

```

`\LWR@closeprevious` $\{ \langle depth \rangle \}$ Close everything up to the given depth:

```

3481 \newcommand*{\LWR@closeprevious}[1]{

```

Close any pending paragraph:

```
3482 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
3483 \whileboolexpr{not test{\ifnumcomp{\LWR@closedepthone}{<}{#1}}}%
3484   {\LWR@closeoneprevious}
3485 }
```

```
3486 \end{warpHTML}
```

41 PDF pages and styles

for HTML output: 3487 \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.

```
3488 \newcommand{\LWR@forcenewpage}{%
3489 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
3490 }
```

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

```
\pagestyle {<style>}
```

```
3491 \renewcommand*{\pagestyle}[1]{}
```

```
\thispagestyle {<style>}
```

```
3492 \renewcommand*{\thispagestyle}[1]{}
```

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

```
3493 \renewcommand*{\markboth}[2]{}
```

```
\markright {<right>}
```

```
3494 \renewcommand*{\markright}[1]{}
```

`\raggedbottom`

3495 `\renewcommand*{\raggedbottom}{}`

`\flushbottom`

3496 `\renewcommand*{\flushbottom}{}`

`\sloppy`

3497 `\renewcommand*{\sloppy}{}`

`\fussy`

3498 `\renewcommand*{\fussy}{}`

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

3499 `\renewcommand*{\pagenumbering}[1]{}`

3500 `\end{warpHTML}`

42 HTML tags, spans, divs, elements

for HTML output: 3501 `\begin{warpHTML}`

42.1 Mapping \TeX Sections to HTML Sections

3502 `\newcommand*{\LWR@tagtitle}{h1}`

3503 `\newcommand*{\LWR@tagtitleend}{/h1}`

3504 `\newcommand*{\LWR@tagpart}{h2}`

3505 `\newcommand*{\LWR@tagpartend}{/h2}`

3506 `\newcommand*{\LWR@tagchapter}{h3}`

3507 `\newcommand*{\LWR@tagchapterend}{/h3}`

3508 `\newcommand*{\LWR@tagsection}{h4}`

3509 `\newcommand*{\LWR@tagsectionend}{/h4}`

3510 `\newcommand*{\LWR@tagsubsection}{h5}`

3511 `\newcommand*{\LWR@tagsubsectionend}{/h5}`

3512 `\newcommand*{\LWR@tagsubsubsection}{h6}`

3513 `\newcommand*{\LWR@tagsubsubsectionend}{/h6}`

3514 `\newcommand*{\LWR@tagparagraph}{span class="paragraph"{}}`

```

3515 \newcommand*\LWR@tagparagraphend}{/span}
3516 \newcommand*\LWR@tag subparagraph}{span class="subparagraph"{} }
3517 \newcommand*\LWR@tag subparagraphend}{/span}
3518
3519 \newcommand*\LWR@tagregularparagraph}{p}

```

42.2 Babel-French

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

(Emulates or patches code by DANIEL FLIPO.)

```

3520 \providecommand*\LWR@FBcancel{}{}
3521
3522 \AtBeginDocument{%
3523 \@ifundefined{frenchbsetup}%
3524 {}%
3525 {%
3526   \frenchbsetup{FrenchFootnotes=false}%
3527 %
3528   \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
3529   \renewcommand*\FBcolonspace{}{%
3530     \begingroup%
3531     \LWR@FBcancel%
3532     \LWR@origampersand{} \nbsp;%
3533     \endgroup%
3534   }%
3535   \renewcommand*\FBthinspace{}{%
3536     \begingroup%
3537     \LWR@FBcancel%
3538     \LWR@origampersand\#x202f;% \,
3539     \endgroup%
3540   }%
3541   \renewcommand*\FBguillspace{}{%
3542     \begingroup%
3543     \LWR@FBcancel%
3544     \LWR@origampersand{} \nbsp;% ~, for \og xyz \fg{}
3545     \endgroup%
3546   }%
3547   \DeclareDocumentCommand\FBmedkern{}{}{%
3548     \begingroup%
3549     \LWR@FBcancel%
3550     \LWR@origampersand\#x202f;% \,
3551     \endgroup%
3552   }%
3553   \DeclareDocumentCommand\FBthickkern{}{}{%

```

```

3554     \begingroup%
3555     \LWR@FBcancel%
3556     \LWR@origampersand{\nbsp;% ~
3557     \endgroup%
3558 }%
3559 \renewcommand*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
3560 \ifFBunicode%
3561 \else%
3562     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
3563     \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
3564 \fi%
3565 }%
3566 }

```

42.3 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.

```

3567 \newcommand*{\LWR@htmltagc}[1]{%
3568 {%
3569 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3570 \begingroup%
3571 \LWR@FBcancel%
3572 \ifmmode\else\protect\LWR@origttfamily\fi%
3573 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3574 \endgroup%
3575 }%
3576 }

```

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.

```

3577 \newcommand*{\LWR@nestspanitem}{%
3578 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
3579 \LWR@origitem%
3580 }
3581
3582 \newenvironment*{LWR@nestspan}
3583 {%
3584 \LWR@traceinfo{LWR@nestspan}%
3585 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%

```

```

3586 {%
3587   \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3588 }%
3589 {% not in a lateximage
3590   \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3591   \addtocounter{LWR@spandepth}{1}%
3592   \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
3593   \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3594   \renewcommand{\BlockClassSingle}[2]{##2}%
3595   \renewcommand{\LWR@forcenewpage}{-}%
3596   \renewcommand{\LWR@liststart}{-}
3597     \let\item\LWR@nestspanitem%
3598   }%
3599   \renewcommand{\LWR@listend}{\LWR@htmltagc{br /}\LWR@htmltagc{br /}}%
3600 }% not in a lateximage
3601 }% starting env
3602 {% ending env
3603 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
3604 }%
3605 {\addtocounter{LWR@spandepth}{-1}}%
3606 \LWR@traceinfo{LWR@nestspan: done}%
3607 }
3608
3609 \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.

```

3610 \NewDocumentCommand{\LWR@htmlspan}{m +m}{-}%
3611 \LWR@ensuredoingapar%
3612 \LWR@htmltagc{#1}%
3613 \begin{LWR@nestspan}%
3614 #2%
3615 \LWR@htmltagc{/#1}%
3616 \end{LWR@nestspan}%
3617 }

```

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

```

3618 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{-}%
3619 \LWR@traceinfo{LWR@htmlspanclass #2}%
3620 \LWR@ensuredoingapar%
3621 \LWR@subhtmlclass{span}[#1]{#2}%
3622 \begin{LWR@nestspan}%
3623 #3%

```

```

3624 \LWR@htmltagc{/span}%
3625 \LWR@traceinfo{LWR@htmlspanclass done}%
3626 \end{LWR@nestspan}%
3627 }

```

`\LWR@htmltag` $\{(\mathit{tag})\}$

Print an HTML tag: `<tag>`

```

3628 \newcommand*{\LWR@htmltagb}[1]{%
3629 \LWR@htmltagc{#1}%
3630 \endgroup%
3631 }
3632
3633 \newcommand*{\LWR@htmltag}{%
3634 \begingroup\catcode'\_ =12
3635 \LWR@htmltagb%
3636 }

```

42.4 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
3637 \newcommand*{\LWR@htmlopencomment}{%
3638 {%
3639 % \LWR@traceinfo{LWR@htmlopencomment}%
3640 \begingroup%
3641 \LWR@FBcancel%
3642 \ifmmode\else\protect\LWR@origttfamily\fi%
3643 \LWR@origtextless{!{-}{-}%
3644 \endgroup%
3645 }%
3646 }
3647
3648 \newcommand*{\LWR@htmlclosecomment}{%
3649 {%
3650 % \LWR@traceinfo{LWR@htmlclosecomment}%
3651 \begingroup%
3652 \LWR@FBcancel%
3653 \ifmmode\else\protect\LWR@origttfamily\fi%
3654 {-}{-}\LWR@origtextgreater{}%
3655 \endgroup%
3656 }%

```

3657 }

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

```
3658 \newcommand{\LWR@htmlcomment}[1]{%
3659 \LWR@htmlopencomment{}}%
3660 {%
3661 \LWR@origttfamily% break ligatures
3662 #1%
3663 }%
3664 \LWR@htmlclosecomment{}}
```

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

```
3665 \newcommand{\LWR@htmlblockcommentb}[1]
3666 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3667
3668 \newcommand{\LWR@htmlblockcomment}
3669 {%
3670 \begingroup\catcode'\_ =12
3671 \LWR@htmlblockcommentb%
3672 }
```

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

```
3673 \newcommand*\LWR@htmlblocktag[1]{%
3674 \LWR@stoppars%
3675 \LWR@htmltag{#1}%
3676 \LWR@startpars%
3677 }
```

42.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```
3678 \NewDocumentCommand{\LWR@subhtmlclass}{m O{} m}{%
3679 \LWR@traceinfo{\LWR@subhtmlclass #1 #3}%
3680 \ifblank{#2}%
3681 {\LWR@htmltag{#1 class="#3"}}% empty option
3682 {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
```

```
3683 \LWR@traceinfo{\LWR@subhtmlclass done}%
3684 }
```

`\LWR@htmlclass` $\langle element \rangle$ $\langle class \rangle$ [$\langle style \rangle$]

```
3685 \NewDocumentCommand{\LWR@htmlclass}{m o m}{%
3686 \LWR@stoppars%
3687 \LWR@subhtmlclass{#1}[#2]{#3}%
3688 \LWR@startpars%
3689 }
```

`\LWR@htmlclassend` $\langle element \rangle$ $\langle class \rangle$

```
3690 \newcommand*{\LWR@htmlclassend}[2]{%
3691 \LWR@stoppars%
3692 \LWR@htmltag{/#1}%
3693 \ifbool{HTMLDebugComments}{%
3694   \LWR@htmlcomment{End of #1 ‘#2’}%
3695 }{}%
3696 \LWR@startpars%
3697 }
```

`\LWR@htmldivclass` [$\langle style \rangle$] $\langle class \rangle$

```
3698 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3699 \LWR@htmlclass{div}[#1]{#2}%
3700 }
```

`\LWR@htmldivclassend` $\langle class \rangle$

```
3701 \newcommand*{\LWR@htmldivclassend}[1]{%
3702 \LWR@htmlclassend{div}{#1}%
3703 }
```

42.6 Single-line elements

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

`\LWR@htmlclassline` $\langle element \rangle$ [$\langle style \rangle$] $\langle class \rangle$ $\langle text \rangle$

```
3704 \NewDocumentCommand{\LWR@htmlclassline}{m o m +m}{%
3705 \LWR@stoppars
3706 \LWR@subhtmlclass{#1}[#2]{#3}%

```

```

3707 #4%
3708 \LWR@htmltag{/#1}
3709 \LWR@startpars
3710 }

```

42.7 HTML5 semantic elements

`\LWR@htmlelement` $\{ \langle element \rangle \}$

```

3711 \newcommand*\LWR@htmlelement}[1]{%
3712 \LWR@htmlblocktag{#1}
3713 }

```

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

```

3714 \newcommand*\LWR@htmlelementend}[1]{%
3715 \LWR@stoppars
3716 \LWR@htmltag{/#1}
3717 \LWR@startpars
3718 }
3719
3720 \end{warpHTML}

```

42.8 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 [77](#).

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

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

for HTML output:

```

3721 \begin{warpHTML}
3722 \NewDocumentEnvironment{BlockClass}{o m}%
3723 {
3724 \LWR@htmldivclass[#1]{#2}
3725 }
3726 {
3727 \LWR@htmldivclassend{#2}

```

```
3728 }
3729 \end{warpHTML}
```

for PRINT output: 3730 \begin{warpprint}
3731 \NewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3732 \end{warpprint}

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

for HTML output: 3733 \begin{warpHTML}
3734 \newcommand{\BlockClassSingle}[2]{%
3735 \LWR@html@elementclassline{div}{#1}{#2}%
3736 }
3737 \end{warpHTML}

for PRINT output: 3738 \begin{warpprint}
3739 \newcommand{\BlockClassSingle}[2]{#2}
3740 \end{warpprint}

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

for HTML output: 3741 \begin{warpHTML}
3742 \NewDocumentCommand{\InlineClass}{o m +m}{%
3743 \LWR@html@spanclass[#1]{#2}{#3}%
3744 }
3745 \end{warpHTML}

for PRINT output: 3746 \begin{warpprint}
3747 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
3748 \end{warpprint}

Env `LWR@BlockClassWP` `{<WPstyle>}{<HTMLstyle>}{<class>}` Low-level interface for <div> classes with an automatic float ID. These are often used when `\ifbool{FormatWP}`.

for HTML output: 3749 \begin{warpHTML}
3750 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{%
3751 {%
3752 \LWR@stoppars%
3753 \ifbool{FormatWP}%
3754 {%
3755 \LWR@maybeinthisautoid%
3756 \LWR@htmltag{%
3757 div class="#3" %
3758 id="autoid-\arabic{LWR@thisautoid}"%
3759 \ifblank{#1}{-}{ style="#1"}%
3760 }%
3761 }% FormatWP
3762 {% not FormatWP

```

3763 \LWR@htmltag{%
3764     div class="#3"%
3765     \ifblank{#2}{#{ style="#2"}%
3766 }%
3767 }% not FormatWP
3768 \LWR@startpars%
3769 }
3770 {\LWR@htmldivclassend{#3}}
3771 \end{warpHTML}

```

for PRINT output:

```

3772 \begin{warpprint}
3773 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{-}{-}%
3774 \end{warpprint}

```

42.9 Closing HTML tags

for HTML output:

```

3775 \begin{warpHTML}

```

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

```

3776 \newcommand*{\LWR@printclosepart}
3777     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3778 \newcommand*{\LWR@printclosechapter}
3779     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3780 \newcommand*{\LWR@printclosesection}
3781     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3782 \newcommand*{\LWR@printclosesubsection}
3783     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3784 \newcommand*{\LWR@printclosesubsubsection}
3785     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3786 \newcommand*{\LWR@printcloseparagraph}
3787     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3788 \newcommand*{\LWR@printclosesubparagraph}
3789     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

3790 \newcommand*{\LWR@printcloselistitem}
3791     {\LWR@htmltag{/li}}
3792 \newcommand*{\LWR@printclosedescitem}
3793     {\LWR@htmltag{/dd}}
3794 \newcommand*{\LWR@printcloseitemize}
3795     {\LWR@htmltag{/ul}}
3796 \newcommand*{\LWR@printcloseenumerate}
3797     {\LWR@htmltag{/ol}}
3798 \newcommand*{\LWR@printclosedescription}

```

```
3799     {\LWR@htmltag{/dl}}
```

```
3800 \end{warpHTML}
```

43 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 44 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: 3801 `\begin{warpHTML}`

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

```
3802 \newcounter{LWR@spandepth}
3803 \setcounter{LWR@spandepth}{0}
```

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

```
3804 \newbool{LWR@doingstartpars}
3805 \boolfalse{LWR@doingstartpars}
```

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

```
3806 \newbool{LWR@doingapar}
3807 \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:

```
3808 \newcommand*{\LWR@ensuredoingapar}{%
3809 \ifbool{LWR@doingstartpars}%
3810 {\global\booltrue{LWR@doingapar}}%
3811 {}}%
3812 }
```

`\LWR@openparagraph`

```
3813 \newcommand*{\LWR@openparagraph}
3814 {%
```

See if paragraph handling is enabled:

```
3815 \ifbool{LWR@doingstartpars}%
3816 {% handling pars
```

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

```
3817   \ifboolexpr{
3818       test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
3819       test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
3820   }% 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 .

```
3821   }% no nested par tags
```

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

```
3822   {% yes nest par tags
3823       \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Manually indent item list labels to avoid left margin intrusion:

See if are nested inside an item list:

```
3824       \ifnumcomp{\@listdepth}{>}{0}%
3825       {%
```

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

```
3826         \LWR@origspace{1in}%
3827         }%
3828         }%
```

Now have started a paragraph.

```
3829         \global\booltrue{LWR@doingapar}%
```

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

```

3830     \let\par\LWR@closeparagraph%
3831   }% end of yes nest par tags
3832}% end of handling pars
3833 {}% not handling pars
3834 }

```

\LWR@closeparagraph

```

3835 \newcommand*{\LWR@closeparagraph}
3836 {%

```

See if paragraph handling is enabled:

```

3837 \ifbool{LWR@doingapar}%

```

If currently in paragraph mode:

```

3838 {% handling pars

```

See if already started a lateximage or a :

```

3839   \ifboolexpr{
3840     test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
3841     test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
3842   }%

```

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

```

3843   {% no nested par tags
3844     \ifboolexpr{
3845       test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}} and
3846       test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
3847     }%
3848     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3849     {}%
3850   }% no nested par tags

```

If have not already started a lateximage or a :

```

3851   {% yes nest par tags

```

Print a closing tag and some extra vertical space:

```

3852      \unskip%
3853      \LWR@htmltagc{/LWR@tagregularparagraph}%
3854      \LWR@orignewline%

```

No longer doing a paragraph:

```

3855      \global\boolfalse{LWR@doingapar}%
3856 % Disable the special \env{minipage} \& \cs{hspace} interaction
3857 % until a new minipage is found:
3858 %   \begin{macrocode}
3859      \global\boolfalse{LWR@minipagethispar}%
3860   }% end of yes nest par tags
3861 }% end of handling pars

```

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

```

3862 {% not handling pars
3863   \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
3864   {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3865   }%
3866 }% not handling pars

```

Finish with regular paragraph processing

```

3867 \LWR@origpar%
3868 }

3869 \end{warpHTML}

```

44 Paragraph start/stop handling

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

Section 43 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: 3870 `\begin{warpHTML}`

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

```

3871 \newcommand*{\LWR@startpars}%
3872 {%

```

Ignore if inside a span:

```
3873 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
3874 {}%
3875 {%
```

See if currently handling HTML paragraphs:

```
3876 \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3877 {}%
```

If not currently in paragraph mode:

```
3878 {%
```

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

```
3879 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3880 \let\par\LWR@closeparagraph
3881
3882 }% an intentionally blank line
```

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

```
3883 \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
3884 \global\boolfalse{LWR@doingapar}%
3885 }% nestspan
3886 }
```

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

```
3887 \newcommand*{\LWR@stoppars}%
3888 {%
```

Ignore if inside a span:

```

3889 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
3890 {}%
3891 {%

```

See if currently handling HTML paragraphs:

```

3892   \ifbool{LWR@doingapar}%

```

if currently in an HTML paragraph:

```

3893   {%

```

Print a closing tag:

```

3894       \unskip%
3895       \LWR@htmltagc{/\LWR@tagregularparagraph}%
3896       \LWR@orignewline%

```

No longer have an open HTML paragraph:

```

3897       \global\boolfalse{LWR@doingapar}%

```

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

```

3898       \global\boolfalse{LWR@minipagethispar}
3899
3900   }% an intentionally blank line

```

If was not in an HTML paragraph:

```

3901   {}%

```

See if currently allowing HTML paragraphs:

```

3902   \ifbool{LWR@doingstartpars}%

```

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

```

3903   {\ClearPreHook{par}}%

```

Else: do nothing

```

3904   {}%

```

no longer in paragraph mode

```

3905   \global\setbool{LWR@doingstartpars}{false}%

```

no <p> tag to undo:

```
3906 \global\boolfalse{LWR@doingapar}%
3907 }% nestspan
3908 }

3909 \end{warpHTML}
```

45 Page headers and footers

for HTML & PRINT: 3910 \begin{warpall}

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

```
3911 \newcommand{\LWR@firstpagetop}{} % for the home page alone
3912 \newcommand{\LWR@pagetop}{} % for all other pages
3913 \newcommand{\LWR@pagebottom}{}
3914
3915 \newcommand{\LWR@setfirstpagetopb}[1]{%
3916 \renewcommand{\LWR@firstpagetop}{#1}
3917 \catcode'\_ =8
3918 }
```

\HTMLFirstPageTop {<text and logos>}

```
3919 \newcommand{\HTMLFirstPageTop}{%
3920 \catcode'\_ =12
3921 \LWR@setfirstpagetopb
3922 }
```

```
3923 \newcommand{\LWR@setpagetopb}[1]{%
3924 \renewcommand{\LWR@pagetop}{#1}
3925 \catcode'\_ =8
3926 }
```

\HTMLPageTop {<text and logos>}

```
3927 \newcommand{\HTMLPageTop}{%
3928 \catcode'\_ =12
3929 \LWR@setpagetopb
3930 }
```

```
3931 \newcommand{\LWR@setpagebottomb}[1]{%
```

```

3932 \renewcommand{\LWR@pagebottom}{#1}
3933 \catcode'\_ =8
3934 }

```

`\HTMLPageBottom` $\{ \langle \textit>text and logos \rangle \}$

```

3935 \newcommand{\HTMLPageBottom}{%
3936 \catcode'\_ =12
3937 \LWR@setpagebottomb
3938 }

3939 \end{warpall}

```

46 CSS

for HTML output: 3940 `\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.

```
3941 \newcommand*{\LWR@currentcss}{lwarp.css}
```

`\CSSFilename` $\{ \langle \textit>new-css-filename.css \rangle \}$ Assigns the CSS file to be used by the following HTML pages.

```

3942 \newcommand*{\LWR@newcssb}[1]{%
3943 \renewcommand*{\LWR@currentcss}{#1}
3944 \catcode'\_ =8
3945 }
3946
3947 \newcommand*{\CSSFilename}{
3948 \catcode'\_ =12
3949 \LWR@newcssb
3950 }
3951 \end{warpHTML}

```

for PRINT output: 3952 `\begin{warpprint}`
 3953 `\newcommand*{\CSSFilename}[1]{}`
 3954 `\end{warpprint}`

47 Title, HTML meta author, HTML meta description

for HTML output: 3955 `\begin{warpHTML}`

`\title` $\langle title \rangle$ Modified to remember `\thetitle`, which is used to set the HTML page titles.

```
3956 \let\LWR@origtitle\title
3957
3958 \renewcommand*{\title}[1]{%
3959     \LWR@origtitle{#1}%
3960     \begingroup%
3961         \renewcommand{\thanks}[1]{}%
3962         \protected@xdef\thetitle{#1}%
3963     \endgroup%
3964 }
```

```
3965 \end{warpHTML}
```

for HTML & PRINT: 3966 `\begin{warpall}`

`\HTMLTitle` $\langle Titlename \rangle$ The Title to place into an HTML meta tag. The default is to use the document `\title`'s setting.

```
3967 \providecommand{\thetitle}{}
3968
3969 \newcommand{\theHTMLTitle}{\thetitle}
3970
3971 \newcommand{\HTMLTitle}[1]{\renewcommand{\theHTMLTitle}{#1}}
```

`\HTMLAuthor` $\langle authorname \rangle$ The author to place into an HTML meta tag. If none given, the default is `\theauthor`, which is empty unless the titling package is used.

```
3972 \providecommand{\theauthor}{}
3973
3974 \newcommand{\theHTMLAuthor}{\theauthor}
3975
3976 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
```

This is placed inside an HTML meta tag at the start of each file. This may be changed mid-document using `\HTMLDescription`, allowing different HTML descriptions to be used for different sections of the document.



Do not use double quotes, and do not exceed 150 characters.

`\HTMLDescription` `{\langleNew HTML meta description.\rangle}` Assigns the HTML file's description meta tag.

```

3977 \newcommand{\LWR@currentHTMLDescription}{}
3978
3979 \newcommand{\HTMLDescription}[1]{%
3980 \renewcommand{\LWR@currentHTMLDescription}{#1}
3981 }
3982
3983 \end{warpall}

```

48 Footnotes

lwarp uses native \TeX footnote code, although with its own `\box` to avoid the \TeX 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.

48.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page using the \TeX 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 48.4 for the implementation.

48.2 Minipage footnotes

See section 48.5 for how minipage footnotes are gathered. See section 76.3 for how minipage footnotes are placed into the document.

48.3 Titlepage thanks

See section 55.6 for titlepage footnotes.

48.4 Regular page footnote implementation

for HTML & PRINT: 3984 `\begin{warpall}`

Ctrl FootnoteDepth Determines how deeply to place footnotes in the HTML files, similar to `tocdepth`.
 Default: 5 The default of 5 places footnotes before each `\subparagraph` or higher. See table 7 for a table of \TeX section headings.

```
3985 \newcounter{FootnoteDepth}
3986 \setcounter{FootnoteDepth}{5}
```

```
3987 \end{warpall}
```

for HTML output: 3988 `\begin{warpHTML}`

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

```
3989 \newbox\LWR@footnotes
```

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

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

```
3990 \long\def\@makefntext#1{\textsuperscript{\@thefnmark} #1}
```

```
\@makefnmark
```

```
3991 \def\@makefnmark{\hbox{\textsuperscript{\@thefnmark}}}
```

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

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

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

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

```
3992 \long\def\LWR@footnotetext#1{%
3993 \LWR@traceinfo{LWR@footnotetext}%
3994 \global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
3995 \unvbox\LWR@footnotes%
```

Remember the footnote number for \ref:

```
3996 \protected@edef\@currentlabel{%
3997 \csname p@footnote\endcsname\@thefnmark%
3998 }% @currentlabel
```

Open a group:

```
3999 \color@begingroup%
```

Use HTML superscripts even inside a lateximage:

```
4000 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

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

```
4001 \ifthenelse{%
4002 \boolean{LWR@doingstartpars} \AND%
4003 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
4004 }%
4005 {%}
4006 {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
4007 \@makefntext{#1}%
```

Closing paragraph tag:

```
4008 \ifthenelse{%
4009 \boolean{LWR@doingstartpars} \AND%
4010 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
4011 }%
4012 {\par}%
4013 {%}
4014 \LWR@htmltagc{/\LWR@tagregularparagraph}%
4015 \LWR@orignewline%
4016 }%
```

Close the group:

```
4017 \color@endgroup%
4018 }% vbox
4019 }%
```

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

```
4020 \let\@footnotetext\LWR@footnotetext
```

48.5 Minipage footnote implementation

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

```
4021 \long\def\@mpfootnotetext#1{%
4022 \global\setbox\@mpfootins\vbox{%
4023   \unvbox\@mpfootins
4024   \reset@font\footnotesize
4025   \hsize\columnwidth
4026   \@parboxrestore
4027   \protected@edef\@currentlabel
4028   {\csname p@mpfootnote\endcsname\@thefnmark}%
4029   \color@begingroup
```

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

```
4030   \ifthenelse{%
4031     \boolean{LWR@doingstartpars} \AND%
4032     \cntttest{\value{LWR@lateximagedepth}}{=}{0}}%
4033   }%
4034   {}%
4035   {\LWR@htmltagc{\LWR@tagregularparagraph}}%

4036   \@makefntext{%
4037     \ignorespaces#1%
4038   }%
```

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

```
4039   \ifthenelse{\cntttest{\value{LWR@lateximagedepth}}{>}{0}}%
4040   }%
4041   {%
4042     \LWR@htmltagc{/\LWR@tagregularparagraph}%
4043     \LWR@orignewline%
4044   }%
4045   \color@endgroup%
4046 }% vbox
4047 }
```

48.6 Printing pending footnotes

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

```
4048 \newcommand*\LWR@printpendingfootnotes}{%
4049 \ifvoid\LWR@footnotes\else
4050   \LWR@forcenewpage
4051   \begin{BlockClass}{footnotes}
4052   \LWR@origmedskip
4053   \unvbox\LWR@footnotes
4054   \setbox\LWR@footnotes=\vbox{}
4055   \end{BlockClass}
4056 \fi
4057 }
```

`\LWR@maybeprintpendingfootnotes` [*depth*] Used to print footnotes before sections only if formatting for an EPUB or word processor:

```
4058 \newcommand*\LWR@maybeprintpendingfootnotes[1]{%
4059 \ifboolexpr{
4060   not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
4061   bool{FormatEPUB} or
4062   bool{FormatWP}
4063 }%
4064 {\LWR@printpendingfootnotes}%
4065 {}%
4066 }

4067 \end{warpHTML}
```

49 Marginpars

`\marginpar` [*left*] [*right*] `\marginpar` may contains paragraphs, but in order to remain inline with the surrounding text lwarp nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `
` tags.

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

for HTML output: 4068 `\begin{warpHTML}`

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

```

4069 \renewcommand{\marginpar}[2] [] {%
4070 \ifbool{FormatWP}%
4071 {%
4072 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{\marginblock}
4073 #2
4074 \end{LWR@BlockClassWP}
4075 }%
4076 {%
4077 \LWR@htmlspanclass{\marginpar}{#2}%
4078 }%
4079 }

```

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

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

HTML version.

```

4080 \newcommand{\marginparBlock}[2] [] {%
4081 \ifbool{FormatWP}%
4082 {%
4083 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{\marginblock}
4084 #2
4085 \end{LWR@BlockClassWP}
4086 }%
4087 {%
4088 \begin{BlockClass}[width:2in; float:right; margin:10pt]{\marginparblock}
4089 #2
4090 \end{BlockClass}
4091 }%
4092 }

4093 \end{warpHTML}

```

for PRINT output: 4094 \begin{warpprint}

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

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

Print version.

```

4095 \LetLtxMacro\marginparBlock\marginpar

4096 \end{warpprint}

```

50 Splitting HTML files

- Files are split according to `FileDepth` and `CombineHigherDepths`.
- Filenames are sanitized by `\LWR@filenamoblanks`.
- `\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: 4097 `\begin{warpall}`

`Ctr FileDepth` `{\langle section depth \rangle}` determines how deeply to break into new HTML files, similar to `tocdepth`. The default of `-5` produces one large HTML file.

```
4098 \newcounter{FileDepth}
4099 \setcounter{FileDepth}{-5}
```

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

```
4100 \newbool{CombineHigherDepths}
4101 \booltrue{CombineHigherDepths}
```

```
4102 \end{warpall}
```

for HTML output: 4103 `\begin{warpHTML}`

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

```
4104 \newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
4105 \newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenamoblanks` `{\langle filename \rangle}`

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.

```
4106 \newcommand*{\LWR@filenamoblanks}[1]{%
4107 \begingroup
```

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

```
4108 \LWR@nullfonts
4109 \renewcommand*{\LWR@htmltagc}[1]{%
```

Replaces common symbols and short words with hyphens:

```
4110 \edef\LWR@thisnewfilename{#1}
4111 \fullexpandarg
```

Convert spaces into hyphens:

```
4112 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
4113 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4114 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
4115 \StrSubstitute{\LWR@thisnewfilename}%
4116   {\LWR@origampersand}{-}[\LWR@thisnewfilename]
4117 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
4118 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4119 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
4120 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
4121 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
4122 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
4123 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
4124 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
4125 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
4126 \StrSubstitute{\LWR@thisnewfilename}%
4127   {\textless}{-}[\LWR@thisnewfilename]
4128 \StrSubstitute{\LWR@thisnewfilename}%
4129   {\textgreater}{-}[\LWR@thisnewfilename]
4130 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
4131 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4132 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4133 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4134 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
4135 \StrSubstitute{\LWR@thisnewfilename}%
4136   {\textbackslash}{-}[\LWR@thisnewfilename]
4137 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4138 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4139 %   "~}" for babel
4140 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4141 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4142 \StrSubstitute{\LWR@thisnewfilename}{' }{-}[\LWR@thisnewfilename]
```

Convert short words:

```

4143 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
4144 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
4145 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
4146 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
4147 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
4148 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
4149 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
4150 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
4151 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
4152 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
4153 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
4154 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
4155 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
4156 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
4157 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
4158 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
4159 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
4160 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

4161 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
4162 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
4163 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
4164 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4165 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4166 % emdash
4167 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4168 % endash
4169 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
4170 \endgroup
4171 }

```

`\LWR@newhtmlfile` $\{ \langle section name \rangle \}$

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.

```
4172 \newcommand*{\LWR@newhtmlfile}[1]{
```

At the bottom of the ending file:

```

4173 \LWR@htmlend{section}{tbody}
4174
4175 \LWR@printpendingfootnotes
4176

```

No footer between files if EPUB:

```
4177 \ifbool{FormatEPUB}
4178 {}
4179 {
4180   \LWR@htmlelement{footer}
4181
4182   \LWR@pagebottom
4183
4184   \LWR@htmlelementend{footer}
4185 }
```

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

```
4186 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4187 {}
4188 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
4189 \LWR@stoppars
4190 \LWR@htmltag{/body}\LWR@orignewline
4191 \LWR@htmltag{/html}\LWR@orignewline
4192 \LWR@orignewpage
4193
4194 \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.

```
4195 \ifbool{FileSectionNames}%
4196 {\LWR@filenamenooblanks{#1}}
4197 {\renewcommand*\LWR@thisfilename{\arabic{LWR@htmlfilenumber}}}
```

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

```
4198 \LWR@htmlblockcomment{%
4199 |Start file|%
4200 \LWR@htmlsectionfilename{\LWR@thisfilename}|%
4201 }
```

At the top of the starting file:

```
4202 \LWR@stoppars
```

4203

```
4204 \LWR@filestart{ -- #1}% there is an EMDash in front of the #1
4205
```

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

```
4206 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4207 {}
4208 {\LWR@topnavigation}
4209
```

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

```
4210 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4211 {}
4212 {
4213   \LWR@html element{header}
4214
4215   \LWR@pagetop
4216
4217   \LWR@html elementend{header}
4218 }
4219
```

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

```
4220 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4221 {}
4222 {\ifcvoid{thetitle}{}\LWR@printthetitle}
4223
```

No sidetoc if formatting for an EPUB or word processor:

```
4224 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4225 {}
4226 {\LWR@sidetoc}
4227
```

Start of the <textbody>:

```
4228 \LWR@html elementclass{section}{textbody}
4229
```

Keep paragraph tags disabled for now:

```
4230 \LWR@stoppars
4231
```

Track the page numbers:

```
4232 \setcounter{LWR@latestautopage}{\value{page}}
4233 }

4234 \end{warpHTML}
```

51 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 \TeX` to generate characters and `pdftotext` to read them. If extra symbols appear in the text, it may be that `pdf \TeX` is actually producing a symbol over or under a character, resulting in `pdftotext` picking up the accent symbol separately.



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

for HTML output: 4235 `\begin{warpHTML}`

51.1 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.

```
4236 \newbool{LWR@forcinghtmlpage}
4237 \boolfalse{LWR@forcinghtmlpage}
4238
4239 \newcommand*{\ForceHTMLPage}{%
4240 \global\booltrue{LWR@forcinghtmlpage}%
4241 }
```

`\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.

```
4242 \newbool{LWR@forcinghtmltoc}
4243 \boolfalse{LWR@forcinghtmltoc}
4244
4245 \newcommand*{\ForceHTMLTOC}{%
4246 \global\booltrue{LWR@forcinghtmltoc}%
4247 }
```

```
4248 \end{warpHTML}
```

for PRINT output:

```
4249 \begin{warpprint}
4250 \newcommand*{\ForceHTMLPage}{}
4251 \newcommand*{\ForceHTMLTOC}{}
4252 \end{warpprint}
```

for HTML output:

```
4253 \begin{warpHTML}
```

51.2 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.

```
4254 \newbool{LWR@mainmatter}
4255 \DeclareDocumentCommand{\mainmatter}{}{}%
4256 \booltrue{LWR@mainmatter}%
4257 }
```

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

```
4258 \DeclareDocumentCommand{\frontmatter}{}{}%
4259 \boolfalse{LWR@mainmatter}%
4260 }
```

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

```
4261 \DeclareDocumentCommand{\backmatter}{}{}%
4262 \boolfalse{LWR@mainmatter}
4263 }
```

51.3 Sectioning support macros

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

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

```
4264 \newcommand*\LWR@sectionnumber[1]{%
4265 \InlineClass{sectionnumber}{#1}%
4266 }
```

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

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

Create an autosection tag.

```
4267 \newcommand*\LWR@createautosec[1]{%
4268 \LWR@htmltag{#1 id="autosec-\thepage"{}%
4269 }
```

`\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.

```
4270 \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.

```
4271 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
4272 \LWR@closeprevious{#1}%
```

Push a new section depth:

```
4273 \LWR@pushoneclose{#1}{#2}%
4274 }
```

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.

```
4275 \newcounter{LWR@prevFileDepth}
4276 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\@secntformat` $\{\langle sectiontype \rangle\}$

```
4277 \def\@secntformat#1{\csname the#1\endcsname\protect\quad}
```

`\simplechapterdelim` Used by tocbibind and anonchap.

```
4278 \newcommand*\simplechapterdelim{\}
```

`\@chaptformat` $\{\langle sectiontype \rangle\}$

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

```
4279 \let\@chaptformat\@secntformat
```

`Ctr LWR@currentautosec` Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding `lateximage` will cause the page number to change by the time the following `autosec` label is created.

```
4280 \newcounter{LWR@currentautosec}
```

`\LWR@section` * $[\langle TOC name \rangle] \{\langle name \rangle\} \{\langle sectiontype \rangle\}$

The common actions for the high-level sectioning commands.

```
4281 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
4282 \LWR@traceinfo{LWR@section |#2| |#3|}%
4283 \LWR@traceinfo{LWR@section: not an empty section}%
4284 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
4285 \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:

```

4286 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4287 \IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{}%
4288 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}%
4289 \ifthenelse{%
4290   \(%
4291     \(\NOT\equal{#1}{\BooleanTrue})\OR%
4292     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart})\OR%
4293     \(\boolean{LWR@forcinghtmlpage})\)%
4294   \)%
4295   \AND%
4296   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
4297   \AND%
4298   \(%
4299     \NOT\boolean{CombineHigherDepths}\OR%
4300     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
4301   \)%
4302   \AND%

4303   \NOT\isempty{#3}% phantomsection
4304 }%

```

If so: start a new HTML file:

```

4305 {% new file
4306   \LWR@traceinfo{LWR@section: new HTML file}%

```

See if there was an optional TOC name entry:

```

4307   \IfNoValueTF{#2}%

```

If no optional entry

```

4308   {\LWR@newhtmlfile{#3}}%

```

If yes an optional entry

```

4309   {\LWR@newhtmlfile{#2}}%
4310 }% new file

```

Else: No new HTML file:

```

4311 {% not new file

```

Generate a new \LaTeX page so that toc and index page number points to the section:

```

4312   \LWR@traceinfo{LWR@section: not a new HTML file}%

```

```
4313 \LWR@orignewpage%
4314
4315 }% not new file
```

Remember this section's name for \nameref:

```
4316 \IfValueT{#3}{%
4317 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4318 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4319 }%
```

Print an opening comment with the level and the name; ex: "section" "Introduction"

```
4320
4321 \ifbool{HTMLDebugComments}{%
4322 \begingroup%
4323 \LWR@nullfonts%
4324 \LWR@htmlcomment{Opening #4 ‘‘#3’’}}%
4325 \endgroup%
4326 }{}%
4327
```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
4328 \ifthenelse{%
4329 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4330 }%
4331 {\LWR@startpars}%
4332 {}%
```

Create the opening tag with an autosec:

```
4333 \LWR@createautosec{\csuse{LWR>tag#4}}%

4334 \setcounter{LWR@currentautosec}{\value{page}}
```

Check if starred:

```
4335 \IfBooleanTF{#1}%
4336 {%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
4337 \ifbool{LWR@forcinghtmltoc}%
4338 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4339 {}%
4340 }% starred
```

Not starred, so step counter and add to TOC:

```
4341 {% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
4342 \ifthenelse{%
4343   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4344 }%
4345 {% if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For article class, lwarp assumes that all is mainmatter.

```
4346 \LWR@traceinfo{LWR@section: about to test main matter}%
4347 \ifbool{LWR@mainmatter}%
4348 {%
4349   \LWR@traceinfo{LWR@section: yes mainmatter}%
4350   \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
4351 \LWR@traceinfo{LWR@section: about to addcontentsline}%
4352 \addcontentsline{toc}{#4}%
4353 {%
4354   \protect\numberline{\csuse{the#4}}%
4355   {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4356 }%
4357 \LWR@traceinfo{LWR@section: finished addcontentsline}%
4358 }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
4359 {% not main matter
4360   \LWR@traceinfo{LWR@section: no main matter}%
4361   \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4362 }% end of not main matter
4363 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
4364 {%
4365   \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4366 }%
```

For part, print the section type:

```
4367 \ifbool{LWR@mainmatter}%
```

```

4368   {%
4369     \ifthenelse{%
4370       \(\cntttest{\csuse{LWR@depth#4}}{<=} %
4371         {\value{secnumdepth}}\)\ \AND%
4372       \(\cntttest{\csuse{LWR@depth#4}}{<=} {\LWR@depthpart}\)\ %
4373     }%
4374     {\csuse{#4name}~{}}%
4375     {}%

```

Print the section number:

```

4376     \LWR@traceinfo{LWR@section: about to print section number}%
4377     \ifthenelse{%
4378       \cntttest{\csuse{LWR@depth#4}}{<=} {\value{secnumdepth}}%
4379     }%
4380     {%
4381       \ifstrequal{#4}{chapter}%
4382       {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
4383       {\protect\LWR@sectionnumber{\@seccntformat{#4}}}%
4384     }%
4385     {}%
4386     \LWR@traceinfo{LWR@section: finished print section number}%
4387   }{}%
4388 }% end of not starred

```

Print the section name:

```
4389 #3%
```

Close the heading tag, such as /H2:

```
4390 \LWR@htmltag{\csuse{LWR@tag#4end}}%
```

Generate a \LaTeX label:

```
4391 \label{autopage-\theLWR@currentautosec}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

4392 \ifthenelse{%
4393   \cntttest{\csuse{LWR@depth#4}}{<} {\LWR@depthparagraph}%
4394 }%
4395 {\LWR@startpars}%
4396 {}%

```

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.

```

4397 \ifthenelse{%
4398   \NOT\equal{#1}{\BooleanTrue}\OR%
4399   \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
4400 }%
4401 {% not starred
4402   \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4403 }% not starred
4404 {}%

```

Reset to defaults if not a phantomsection:

```

4405 \ifstrempy{#3}%
4406 {}%
4407 {%
4408 \global\boolfalse{LWR@forcinghtmlpage}%
4409 \global\boolfalse{LWR@forcinghtmltoc}%
4410 }%
4411 %
4412 \LWR@traceinfo{LWR@section: done}%
4413 }

```

51.4 `\section` and friends

`\part` * [*TOC name*] {*name*}

```

4414 \newcommand{\part@preamble}{}% for koma-script
4415
4416 \DeclareDocumentCommand{\part}{s o m}{%
4417 \LWR@maybepreprintpendingfootnotes{\LWR@depthpart}%
4418 \LWR@stoppars%
4419
4420 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4421
4422 \LWR@section{#1}{#2}{#3}{part}%
4423
4424 \part@preamble% for koma-script
4425 \renewcommand{\part@preamble}{}%
4426 }

```

`\chapter` * [*TOC name*] {*name*}

```
4427 \let\@printcites\relax% for quotchap package
4428
4429 \newcommand{\chapter@preamble}{}% for koma-script
4430
4431 \@ifundefined{chapter}
4432 {}
4433 {%
4434 \DeclareDocumentCommand{\chapter}{s o m}{%
4435 \LWR@traceinfo{chapter #3}%
4436 \LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
4437 \LWR@stoppars%
4438
4439 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4440
4441 \LWR@section{#1}{#2}{#3}{chapter}%
4442
4443 \@printcites% for quotchap package
4444
4445 \chapter@preamble% for koma-script
4446 \renewcommand{\chapter@preamble}{}%
4447 }
4448 }
```

`\section` * [*TOC name*] {*name*}

```
4449 \DeclareDocumentCommand{\section}{s o m}{%
4450 \LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
4451 \LWR@stoppars%
4452
4453 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4454
4455 \LWR@section{#1}{#2}{#3}{section}%
4456 }
```

`\subsection` * [*TOC name*] {*name*}

```
4457 \DeclareDocumentCommand{\subsection}{s o m}{%
4458 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
4459 \LWR@stoppars%
4460
4461 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4462
4463 \LWR@section{#1}{#2}{#3}{subsection}%
4464 }
```

`\subsubsection` * [*TOC name*] {*name*}

```
4465 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4466 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
4467 \LWR@stoppars%
4468
4469 \LWR@startnewdepth{\LWR@depthsubsubsection}%
4470 {\LWR@printclosesubsubsection}%
4471
4472 \LWR@section{#1}{#2}{#3}{subsubsection}%
4473 }
```

`\paragraph` * [*TOC name*] {*name*}

```
4474 \DeclareDocumentCommand{\paragraph}{s o m}{%
4475 \LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
4476 \LWR@stoppars%
4477
4478 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4479
4480 \LWR@section{#1}{#2}{#3}{paragraph}%
4481 }
```

`\subparagraph` * [*TOC name*] {*name*}

```
4482 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4483 \LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
4484 \LWR@stoppars%
4485
4486 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4487
4488 \LWR@section{#1}{#2}{#3}{subparagraph}%
4489 }

4490 \end{warpHTML}
```

52 Starting a new file

for HTML & PRINT: 4491 `\begin{warpall}`

`\HTMLLanguage` Default language for the HTML lang tag.

```
4492 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4493
```

```

4494 \newcommand*{\HTMLLanguage}[1]{%
4495 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4496 }

4497 \end{warpall}

```

for HTML output: 4498 \begin{warpHTML}

\LWR@filestart {<title_suffix>}

Creates the opening HTML tags.

```

4499 \newcommand*{\LWR@filestart}[1]{
4500 \LWR@traceinfo{\LWR@filestart}

```

Locally temporarily disable direct-formatting commands:

```

4501 \begingroup
4502 \LWR@traceinfo{\LWR@filestart: A}
4503 \LWR@nullfonts
4504 \LWR@traceinfo{\LWR@filestart: B}

```

Create the page's HTML header:

```

4505 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4506 \LWR@traceinfo{\LWR@filestart: C}

```

The language is user-adjustable:

```

4507 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline

```

Start of the meta data:

```

4508 \LWR@htmltag{head}\LWR@orignewline

```

Charset is fixed at UTF-8:

```

4509 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline

```

Author:

```

4510 \ifthenelse{\equal{\theHTMLAuthor}{}}{
4511 }{
4512 {\LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline}}%

```

lwarp is the generator:

```
4513 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
4514   \LWR@orignewline
```

If there is a description, add it now:

```
4515 \ifdefempty{\LWR@currentHTMLDescription}{-}{-%
4516 \LWR@htmltag{%
4517 meta name="description" content="\LWR@currentHTMLDescription" /}%
4518   \LWR@orignewline
4519 }%
```

Mobile-friendly viewport:

```
4520 \LWR@htmltag{meta name="viewport" %
4521 content="width=device-width, initial-scale=1.0" /}%
4522   \LWR@orignewline
```

IE patch:

```
4523 \LWR@htmltag{!{-}{-}[if lt IE 9]\LWR@orignewline
4524 \LWR@htmltag{%
4525 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"{-}}%
4526 \LWR@htmltag{/script}\LWR@orignewline
4527 \LWR@htmltag{![endif]{-}{-}\LWR@orignewline
```

The page's title:

```
4528 \ifthenelse{\equal{\theHTMLTitle}{}}%
4529 {}%
4530 {\LWR@htmltag{title}\theHTMLTitle#1\LWR@htmltag{/title}\LWR@orignewline}%
```

The page's stylesheet:

```
4531 \LWR@htmltag{%
4532 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4533 \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.

```
4534 \ifbool{mathjax}%
4535 {%
4536   \begingroup%
4537   \LWR@restoreoriglists%
4538   \boolfalse{LWR@verbtags}
4539   \verbatiminput{lwarp_mathjax.txt}%
4540   \booltrue{LWR@verbtags}
```

```

4541   \endgroup%
4542   \LWR@stoppars
4543 }% end of mathjax
4544 {}%

```

End of the header:

```
4545 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```

4546 \LWR@htmltag{body}\LWR@orignewline
4547 \endgroup
4548 \LWR@traceinfo{LWR@filestart: done}
4549 }

```

```
4550 \end{warpHTML}
```

53 Starting HTML output

for HTML output: 4551 \begin{warpHTML}

\LWR@LwarpStart Executed at the beginning of the entire document.

```

4552 \catcode'\$=\active
4553 \newcommand*\LWR@LwarpStart{
4554   {%
4555   \LWR@traceinfo{LWR@lwarpStart}

```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```

4556 \ifbool{FormatWP}{-%
4557   \setcounter{FileDepth}{-5}%
4558   \boolfalse{HTMLDebugComments}%
4559 }{}

```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```

4560 \edef\LWR@strresult{\HomeHTMLFilename}
4561 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4562 \edef\LWR@strresult{\HTMLFilename}
4563 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}

```

Force onecolumn and empty page style:

```
4564 \LWR@origonecolumn%
4565 \LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
4566 \LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
4567 \LWR@origraggedright%
4568 \LetLtxMacro{\}\{\LWR@endofline}%
```

Spread the lines for pdftotext to read them well:

```
4569 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4570 \setlength{\parindent}{0pt}
4571 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
4572 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
4573 \setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
4574 \renewcommand{\ps@plain}{} 
```

```
\centering Not used in the HTML environment:
\raggedleft
\raggedright 4575 \renewcommand*{\centering}{}
4576 \renewcommand*{\raggedleft}{}
4577 \renewcommand*{\raggedright}{} 
```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```
4578 \LetLtxMacro{\LWR@origtabular}{\tabular}
4579 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
```

```
4580 \LetLtxMacro{\tabular}{\LWR@tabular}
4581 \LetLtxMacro{\endtabular}{\endLWR@tabular}
```

Float captions:

```
4582 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

[Label in HTML](#)

```
4583 \let\LWR@origltx@label\ltx@label
4584 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
4585 \let\TeX\LWR@TeX
4586 \let\LaTeX\LWR@LaTeX
4587 \let\LuaTeX\LWR@LuaTeX
4588 \let\LuaLaTeX\LWR@LuaLaTeX
4589 \let\XeTeX\LWR@XeTeX
4590 \let\XeLaTeX\LWR@XeLaTeX
4591 \let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
4592 \global\boolfalse{LWR@doingapar}
4593 \global\boolfalse{LWR@doingstartpars}
```

Start a new HTML file and a header:

```
4594 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4595 \LWR@filestart{}
4596 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4597 \LWR@htmltag{header}\LWR@orignewline
4598 \LWR@startpars
4599 \LWR@firstpagetop
4600 \LWR@stoppars
4601 \LWR@htmltag{/header}\LWR@orignewline
4602 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4603 \LWR@htmltag{section class="textbody"{{}}
```

Document and page settings:

```
4604 \mainmatter
4605 \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`.

```
4606 \LWR@patchlists
```

Ensure that math mode is active to call lwrap's patches:

```
4607 \catcode'\$=\active
```

Allow HTML paragraphs to begin:

```
4608 \LWR@startpars
```

```
4609 \LWR@traceinfo{LWR@lwrapStart: done}
```

```
4610 }
```

```
4611 \catcode'\$=3% math shift until lwrap starts
```

```
4612 \end{warpHTML}
```

54 Ending HTML output

for HTML output: 4613 `\begin{warpHTML}`

`\LWR@requesttoc` `{\boolean}` `{\suffix}` Requests that a toc, lof, or lot be generated.

```
4614 \newcommand*\LWR@requesttoc}[2]{%
```

```
4615 \ifbool{#1}
```

```
4616 {
```

```
4617   \expandafter\newwrite\csuse{tf@#2}
```

```
4618   \immediate\openout \csuse{tf@#2} \jobname.#2\relax
```

```
4619 }{}
```

```
4620 }
```

`\LWR@LwrapEnd` Final stop of all HTML output:

```
4621 \newcommand*\LWR@LwrapEnd
```

```
4622 {
```

```
4623 \LWR@stoppars
```

```
4624 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the `textbody`:

```
4625 \LWR@htmlend{section}{textbody}
```

Print any pending footnotes:

```
4626 \LWR@printpendingfootnotes
```

Create the footer:

```
4627 \LWR@html element{footer}
4628
4629 \LWR@pagebottom
4630
4631 \LWR@html elementend{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.

```
4632 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4633 {}
4634 {
4635   \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{-}
4636 }
```

```
4637 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4638 \LWR@htmltag{/body}\LWR@orignewline
4639 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4640 \LWR@orignewpage
```

For lateximage commands:

```
4641 \immediate\closeout\LWR@lateximagesfile
4642 }
```

```
4643 \end{warpHTML}
```

55 Title page

[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 55.7.
- 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.

55.1 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` or `titlingpage` environments in cases where `\maketitle` is not desired. These commands preserve the `\thanks`, etc., and should not be used in the main text.

`\thetitle` `\thetitle`, `\theauthor`, and `\thedata` are available if titling has been loaded, and are sanitized user-level versions from which have been removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

`\HTMLPageBottom` `{\langle text \rangle}`

```
\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.

55.2 Changes for `\affiliation`

`\affiliation` `{\langle text \rangle}`

Adds the affiliation to the author for use in `\maketitle`.

Inside `titlepage`, this macro prints its argument. Outside, it is null.

for HTML & PRINT:

```
4644 \begin{warpall}
4645 \newrobustcmd{\affiliation}[1]{ }
4646 \end{warpall}
```

```

for PRINT output: 4647 \begin{warpprint}

4648 \AtBeginEnvironment{titlepage}{
4649 \renewrobustcmd{\affiliation}[1]{\ \ \textsc{\small#1}}
4650 }
4651
4652 \AtBeginDocument{
4653 \@ifpackageloaded{titling}{
4654 \AtBeginEnvironment{titlingpage}{
4655 \renewrobustcmd{\affiliation}[1]{\ \ \textsc{\small#1}}
4656 }
4657 }{}% titling loaded
4658 }% AtBeginDocument

4659 \end{warpprint}

```

```

for HTML output: 4660 \begin{warpHTML}

```

Env titlepage Sets up a <div> of class titlepage.

```

4661 \renewenvironment*{titlepage}
4662 {
4663 \renewrobustcmd{\affiliation}[1]{\ \ \InlineClass{affiliation}{##1}}
4664 \LWR@printpendingfootnotes
4665 \LWR@forcenewpage
4666 \BlockClass{titlepage}
4667 }
4668 {
4669 \endBlockClass
4670 \LWR@printpendingfootnotes
4671 }

4672 \end{warpHTML}

```

55.3 Printing the thanks

```

for HTML & PRINT: 4673 \begin{warpall}

```

\printthanks Forces the \thanks to be printed.

This is necessary in a titlingpage environment when \maketitle was not used.

```

4674 \newcommand*{\printthanks}{\@thanks}

4675 \end{warpall}

```

55.4 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: 4676 `\begin{warpHTML}`

`\printtitle`

```
4677 \newcommand*{\printtitle}
4678 {
4679 \LWR@stoppars
4680 \LWR@htmltag{\LWR@tagtitle}%
4681 \@title%
4682 \LWR@htmltag{\LWR@tagtitleend}
4683 \LWR@startpars
4684 }
```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```
4685 \newcommand*{\LWR@printthetitle}
4686 {
4687 \LWR@stoppars
4688 \LWR@htmltag{\LWR@tagtitle}%
4689 \thetitle%
4690 \LWR@htmltag{\LWR@tagtitleend}
4691 \LWR@startpars
4692 }
```

`\printauthor` HTML version.

```
4693 \newcommand*{\printauthor}{
```

The entire author block is contained in a `<div>` named `author`:

```
4694 \begin{BlockClass}{author}
```

`\and` finishes one author and starts the next:

```
4695 \renewcommand{\and}{%
4696 \end{BlockClass}
4697 \begin{BlockClass}{oneauthor}
4698 }
```

Individual authors are contained in a `<div>` named `oneauthor`:

```

4699 \begin{BlockClass}{oneauthor}
4700 \@author
4701 \end{BlockClass}
4702 \end{BlockClass}
4703 }

```

`\printdate`

```

4704 \newcommand*\printdate{%
4705 \begin{BlockClass}{titledate}
4706 \@date
4707 \end{BlockClass}
4708 }

```

```
4709 \end{warpHTML}
```

55.5 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: 4710 `\begin{warpprint}`

`\printtitle`

```
4711 \newcommand*\printtitle{{\Huge\@title}}
```

`\printauthor` Print mode.

```

4712 \newcommand*\printauthor{
4713   {\large\begin{tabular}[t]{c}\@author\end{tabular}}}

```

`\printdate`

```
4714 \newcommand*\printdate{{\small\textit{\@date}}}
```

```
4715 \end{warpprint}
```

55.6 \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 48 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4716 `\begin{warpHTML}`

```
4717 \if@titlepage
4718 \newcommand{\LWR@setfootnoteseries}{%
4719   \renewcommand\thefootnote{\@arabic\c@footnote}%
4720 }
4721 \else
4722 \newcommand{\LWR@setfootnoteseries}{%
4723   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
4724 }
4725 \fi
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
4726 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
4727 \LWR@setfootnoteseries%
4728 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

`\thefootnote` \Rightarrow `\nameuse{arabic}{footnote}`, or
`\thefootnote` \Rightarrow `\nameuse{fnsymbol}{footnote}`

Redefine the footnote text:

```
4729 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4730 \textsuperscript{\@thefnmark} \LWR@origspace{1in}
```

`\makethanksmark` \Rightarrow `\thanksfootmark` \Rightarrow `\tamark` \Rightarrow
`\@thefnmark` \Rightarrow `\itshape a` (or similar)

Print the text:

```
4731 ##1%
4732 }%
4733 }
```

`\@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`.

```
4734 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4735 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
4736 **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
4737 \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.

```
4738 \renewcommand*\maketitle}{%
```

An HTML titlepage `<div>` is used for all classes.

```
4739 \begin{titlepage}
```

Set up special patches:

```
4740 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4741 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4742 \@thanks
```

Close the HTML titlepage div and cleanup:

```
4743 \end{titlepage}
4744 \setcounter{footnote}{0}%
4745 \global\let\thanks\relax
4746 \global\let\maketitle\relax
4747 \global\let\@maketitle\relax
4748 \global\let\@thanks\@empty
4749 \global\let\@author\@empty
4750 \global\let\@date\@empty
4751 \global\let\@title\@empty
4752 \global\let\title\relax
4753 \global\let\author\relax
4754 \global\let\date\relax
4755 \global\let\and\relax
4756 }
```

`\@maketitle` HTML mode. Typesets the title, etc.:

```

4757 \DeclareDocumentCommand{\@maketitle}{-}{%
4758   \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
4759   \@title
4760   \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
4761   \begin{BlockClass}{author}
4762     \renewcommand{\and}{
4763       \end{BlockClass}
4764       \begin{BlockClass}{oneauthor}
4765     }
4766     \begin{BlockClass}{oneauthor}
4767       \@author
4768     \end{BlockClass}
4769   \end{BlockClass}
4770   \begin{BlockClass}{titledate}
4771     \@date
4772   \end{BlockClass}
4773 }
```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```
4774 \newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
4775 \@thanks
```

Set up special patches:

```
4776 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4777 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4778 \@thanks
```

```
4779 }
```

```
4780 \end{warpHTML}
```

55.7 `\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.

△ titling hooks

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. 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: 4781 `\begin{warpall}`

`\AddSubtitlePublished` Adds `\published` and `\subtitle`, and related.

```

4782 \AfterPackage{titling}{
4783 \newcommand*{\AddSubtitlePublished}{%
4784
4785 \newcommand{\@published}{%
4786
4787 \newcommand{\published}[1]{\gdef\@published{##1}}
4788
4789 \renewcommand*{\maketitlehooka}{\printpublished}
4790
4791 \newcommand*{\printpublished}{%
4792 \warpprintonly{\begin{center}\unskip}%
4793 \begin{BlockClass}{published}%
4794 \warpprintonly{\large\itshape}%
4795 \@published%
4796 \end{BlockClass}%
4797 \warpprintonly{\end{center}}}%
4798 }
4799
4800 \newcommand{\@subtitle}{%
4801
4802 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}
4803
4804 \renewcommand*{\maketitlehookb}{\printsubtitle}
4805
4806 \newcommand*{\printsubtitle}{%
4807 \warpprintonly{\begin{center}\unskip}%
4808 \begin{BlockClass}{subtitle}%
4809 \warpprintonly{\Large\itshape}%
4810 \@subtitle%
4811 \end{BlockClass}%
4812 \warpprintonly{\end{center}}}%
4813 }
4814
4815 }% \AddSubtitlePublished

```

```
4816 }% AfterPackage
```

```
4817 \end{warpall}
```

56 Abstract

The following code replaces the \TeX default, and will itself be replaced later if the abstract package is loaded.

for HTML output: 4818 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

Also over-written by the babel package.

```
4819 \providecommand*{\abstractname}{Abstract}
```

Env abstract

```
4820 \DeclareDocumentEnvironment{abstract}{}
4821 {
```

```
4822 \LWR@forcenewpage
```

```
4823 \BlockClass{abstract}
```

```
4824 \BlockClassSingle{abstracttitle}{\abstractname}
```

```
4825 }
```

```
4826 {
```

```
4827 \endBlockClass
```

```
4828 }
```

```
4829 \end{warpHTML}
```

57 Quote and verse

57.1 Citations and attributions

\attribution for use inside quote, quotation, verse:

ex: \attribution{author name} --- \citetitle{book name}

for HTML output: 4830 \begin{warpHTML}

```
4831 \newcommand{\attribution}[1]{%
```

```
4832 \InlineClass{attribution}{--\,#1}}% emdash
4833 \end{warpHTML}
```

for PRINT output:

```
4834 \begin{warpprint}
4835 \newcommand{\attribution}[1]{\textsc{--\,#1}}
4836 \end{warpprint}
```

`\citetitle` for use inside quote, quotation, verse:

for HTML output:

```
4837 \begin{warpHTML}
4838 \newcommand{\citetitle}[1]{%
4839 \InlineClass{citetitle}{--\,#1}}% emdash
4840 \end{warpHTML}
```

for PRINT output:

```
4841 \begin{warpprint}
4842 \newcommand{\citetitle}[1]{\textsl{--\,#1}}
4843 \end{warpprint}
```

57.2 Quotes, quotations

for HTML output:

```
4844 \begin{warpHTML}
```

Env quote

```
4845 \renewenvironment*{quote}
4846 {
4847 \LWR@forcenewpage
4848 \LWR@htmlblocktag{blockquote}
4849 }
4850 {\LWR@htmlblocktag{/blockquote}}
4851
4852 \renewenvironment*{quotation}
4853 {
4854 \LWR@forcenewpage
4855 \LWR@htmlblocktag{blockquotation}
4856 }
4857 {\LWR@htmlblocktag{/blockquotation}}

4858 \end{warpHTML}
```

57.3 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	<code>\leftskip</code>	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 <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 verse 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.
Len	<code>\leftmargini</code>	
Len	<code>\TMLvleftskip</code>	
Len	<code>\TMLleftmargini</code>	

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: 4859 `\begin{warpall}`

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

Len	<code>\TMLvleftskip</code>	Sets <code>\vleftskip</code> inside a verse environment in HTML.
-----	----------------------------	--

```
4860 \newlength{\HTMLvleftskip}
4861 \setlength{\HTMLvleftskip}{1em}
```

Len	<code>\TMLleftmargini</code>	Sets <code>\leftmargini</code> inside a verse environment in HTML.
-----	------------------------------	--

```
4862 \newlength{\HTMLleftmargini}
4863 \setlength{\HTMLleftmargini}{4.5em}
```

```
4864 \end{warpall}
```

58 Verbatim and tabbing

for HTML & PRINT: 4865 `\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.

```
4866 \newlength{\VerbatimHTMLWidth}
4867 \setlength{\VerbatimHTMLWidth}{4in}
4868 \end{warpall}
```

for HTML output: 4869 `\begin{warpHTML}`

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
4870 \newbool{LWR@verbtags}
4871 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [*style*] {*class*}

Encloses a verbatim environment with the given css class.

```
4872 \newcommand*{\LWR@atbeginverbatim}[2] []
4873 {%
```

Avoid excessive space between lines:

```
4874 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4875 \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.

```
4876 \ifbool{LWR@verbtags}{%
4877   \LWR@htmltag{pre class="#2"
4878   \ifthenelse{\equal{#1}{}}{}{style="#1"}%
4879 }% pre
4880 }{)%
```

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.

```
4881 \begingroup%
4882 \LWR@origttfamily%
```

Since inside a `<pre>`, restore the original list processing:

```
4883 \LWR@restoreoriglists%
```

Turn off babel-french extra space before punctuation:

```
4884 \LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a `verse` par. Restore plain \TeX `\hspace` functionality:

```
4885 \LetLtxMacro{\hspace}{\LWR@orighspace}%
4886 }
```

`\LWR@afterendverbatim` Finishes enclosing a `verbatim` environment.

```
4887 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
4888 \endgroup%
4889 \unskip%
```

At the end of the environment, close the `pre`:

```
4890 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}}
4891
4892 }{ }%
```

Resume regular paragraph handling:

```
4893 \LWR@startpars%
4894 }
```

`\verbatiminput` $\{ \langle filename \rangle \}$

Patch `\verbatiminput` to add HTML tags:

```
4895 \let\LWRV@origverbatim@input\verbatim@input
4896
```

```

4897 \renewcommand{\verbatim@input}[2]{%
4898 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
4899 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4900 \LWRV@origverbatim@input{#1}{#2}%
4901 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4902 }

```

Env `verbatim`

```

4903 \AfterEndPreamble{
4904 \LWR@traceinfo{Patching verbatim.}
4905 \AtBeginEnvironment{verbatim}{%
4906 \LWR@forcenewpage
4907 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4908 }
4909 \AfterEndEnvironment{verbatim}{%
4910 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4911 }
4912 }

```

Env `tabbing`

```

4913 \AfterEndPreamble{
4914 \LWR@traceinfo{Patching tabbing.}
4915 \AtBeginEnvironment{tabbing}{%
4916 \LWR@forcenewpage
4917 \LWR@atbeginverbatim{tabbing}\unskip\LWR@origvspace*{-\baselineskip}%
4918 }
4919 \AfterEndEnvironment{tabbing}{%
4920 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4921 }
4922 }

4923 \end{warpHTML}

```

59 Theorems

`\newtheorem` $\{\langle text \rangle\}$ [$\langle counter \rangle$] -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: 4924 `\begin{warpHTML}`

`\@begintheorem` $\langle name \rangle$ $\langle number \rangle$

```
4925 \renewcommand{\@begintheorem}[2]{%
4926 \LWR@forcenewpage
4927 \BlockClass{theoremcontents}
4928 \InlineClass{theoremlabel}{#1\ #2\ }
4929 }
```

`\@opargbegintheorem` $\langle name \rangle$ $\langle number \rangle$ $\langle oparg \rangle$

```
4930 \renewcommand{\@opargbegintheorem}[3]{%
4931 \LWR@forcenewpage
4932 \BlockClass{theoremcontents}
4933 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4934 }
```

`\@endtheorem`

```
4935 \renewcommand*\@endtheorem{%
4936 \endBlockClass% theoremcontents
4937 }
```

```
4938 \end{warpHTML}
```

60 Lists

The environments `itemize`, `enumerate`, and `description` are patched when `lwarp` is started. These patches support the standard \TeX 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 \TeX 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}

```

60.1 List environment

for HTML output: 4939 `\begin{warpHTML}`

The following may be locally redefined by `enumerate` or `description`.

```

4940 \newcommand*{\LWR@printcloseitemize}{\LWR@printcloseitemize}
4941 \newcommand*{\LWR@printopenlist}{ul style="list-style-type:none"}

```

`\LWR@listitem` [*label*]

Handles `\item` inside a list, `itemize`, or `enumerate`.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```

4942 \newcommand*{\LWR@listitem}{%
4943 \LWR@stoppars%
4944 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
4945 \LWR@htmltag{li}%
4946 \LWR@startpars%
4947 \LWR@origitem%
4948 }

```

Env `list` `{label}` `{commands}`

```

4949 \newcommand*{\LWR@liststart}{%
4950 \LWR@stoppars%
4951 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
4952 \LWR@htmltag{\LWR@printopenlist}\LWR@originewline%
4953 \LWR@startpars%
4954 \setlength{\topsep}{0pt}%
4955 \setlength{\partopsep}{0pt}%
4956 \setlength{\itemsep}{0pt}%

```

```

4957 \setlength{\parsep}{0pt}%
4958 \setlength{\leftmargin}{0pt}%
4959 \setlength{\rightmargin}{0pt}%
4960 \setlength{\listparindent}{0pt}%
4961 \setlength{\itemindent}{0pt}%
4962 \setlength{\labelsep}{1em}%
4963 }
4964
4965 \newcommand*\LWR@listend}{%
4966 \LWR@stoppars%
4967 \LWR@closeprevious{\LWR@depthlistitem}%
4968 \LWR@closeoneprevious%
4969 \LWR@startpars%
4970 }

```

60.2 Itemize

`\LWR@itemizeitem` [*<label>*]

Handles `\item` inside an `itemize` or `enumerate`.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```

4971 \newcommand*\LWR@itemizeitem}{%
4972 \LWR@stoppars%
4973 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitem}%
4974 \LWR@htmltag{li}%
4975 \LWR@startpars%
4976 \LWR@origitem%
4977 }

```

Env `itemize` [*<options>*]

```

4978 \newcommand*\LWR@itemizestart}{%
4979 \renewcommand*\LWR@printcloseitem}{\LWR@printcloseitemize}
4980 \renewcommand*\LWR@printopenlist}{ul style="list-style-type:none"}
4981 \let\item\LWR@itemizeitem%
4982 }

```

60.3 Enumerate

An HTML unordered list is used with customized \LaTeX -generated labels.

Env `enumerate` [*options*]

```
4983 \newcommand*\LWR@enumeratestart}{%
4984 \renewcommand*\LWR@printcloselist}{\LWR@printcloseitemize}
4985 \renewcommand*\LWR@printopenlist}{ul style="list-style-type:none"}
4986 \let\item\LWR@itemizeitem%
4987 }
```

60.4 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```
4988 \newcommand*\LWR@descitem}[1][]{%
4989 {%
4990 \LWR@stoppars%
4991 \LWR@setlatestname{#1}%
4992 \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.

```
4993 \LetLtxMacro{\hspace}{\LWR@nohspace}%
```

Process the original `\item` code:

```
4994 \LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```
4995 \LetLtxMacro{\hspace}{\LWR@hspace}%
```

Be sure the label doesn't print to the left of the rest of the file:

```
4996 \LWR@orighspace{1in}
4997 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4998 \LWR@orignewline%
4999 \LWR@htmltag{dd}%
5000 \LWR@startpars%
5001 }
```

Env `description` [*options*]

```
5002 \newcommand*\LWR@descriptionstart}{%
5003 \renewcommand*\LWR@printcloselist}{\LWR@printclosedescription}
5004 \renewcommand*\LWR@printopenlist}{dl}
```

```
5005 \let\item\LWR@descitem%
5006 }
```

60.5 Patching the lists

```
5007 \let\LWR@orig@trivlist\@trivlist
5008 \let\LWR@origtrivlist\trivlist
5009 \let\LWR@origendtrivlist\endtrivlist
```

`\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`.

```
5010 \newcommand*\LWR@patchlists}{%
5011   \let\item\LWR@listitem%
5012   \renewcommand*\@trivlist}{%
5013     \LWR@liststart%
5014     \LWR@orig@trivlist%
5015   }%
5016   \renewcommand*\endtrivlist{\LWR@origendtrivlist\LWR@listend}%
5017   \renewcommand*\trivlist}{%
5018     \LWR@origtrivlist%
5019     \def\makelabel####1{\LWR@origspace{3em}####1}%
5020   }%
5021   \renewcommand*\itemize{\LWR@itemizestart\LWR@origitemize}%
5022   \renewcommand*\enumerate{\LWR@enumeratestart\LWR@origenumerate}%
5023   \renewcommand*\description{\LWR@descriptionstart\LWR@origdescription}%
5024 %   \AtBeginEnvironment{itemize}{\LWR@itemizestart}%
5025 %   \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}%
5026 %   \AtBeginEnvironment{description}{\LWR@descriptionstart}%
5027 }
```

`\LWR@restoreoriglists` Restores the original `trivlist` environment.

```
5028 \newcommand*\LWR@restoreoriglists}{%
5029 \let\@trivlist\LWR@orig@trivlist%
5030 \let\trivlist\LWR@origtrivlist%
5031 \let\endtrivlist\LWR@origendtrivlist%
5032 \LetLtxMacro\item\LWR@origitem%
5033 \LetLtxMacro\itemize\LWR@origitemize%
5034 \LetLtxMacro\enditemize\LWR@endorigitemize%
5035 \LetLtxMacro\enumerate\LWR@origenumerate%
5036 \LetLtxMacro\endenumerate\LWR@endorigenumerate%
```

```

5037 \LetLtxMacro\description\LWR@origdescription%
5038 \LetLtxMacro\enddescription\LWR@endorigdescription%
5039 }

5040 \end{warpHTML}

```

61 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

61.1 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 `\warppHTMLonly` must be used to make versions for print and HTML.
- See section 192.2 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:

```

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

```

⚠ empty cells

`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.

⚠ `\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 \\}
```

61.2 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

for HTML output: 5041 `\begin{warppHTML}`

`\LWR@mynexttoken`

5042 `\newcommand\LWR@mynexttoken\relax`

`\futurelet` copies the next token then executes a function to analyze

`\LWR@futurenonSPACElet` does the same, but ignores intervening white space

Based on the `booktabs` style:

`\LWR@futurenonSPACElet`

```
5043 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
5044 \afterassignment\LWR@fnslone\let\nexttoken= }
5045 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
5046 \def\LWR@fnsltwo{%
5047 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
5048 \else\let\next=\nexttoken\fi\next}
5049 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

5050 `\newcommand*{\LWR@getmynexttoken}{%`

```

5051 \LWR@traceinfo{LWR@getmynexttoken}%
5052 % nothing must follow this next line
5053 \LWR@futurenonpacelet\LWR@mynexttoken\LWR@tabledatacolumnntag
5054 }

```

61.3 Booleans

Bool LWR@startedrow True if should print a row tag before this column.

```

5055 \newbool{LWR@startedrow}
5056 \boolfalse{LWR@startedrow}

```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```

5057 \newbool{LWR@tabularcelladded}
5058 \boolfalse{LWR@tabularcelladded}

```

Bool LWR@doinghline True if the next row will have an hline or midrule above it. Also used for \midrule.

```

5059 \newbool{LWR@doinghline}
5060 \boolfalse{LWR@doinghline}

```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```

5061 \newbool{LWR@doingtbrule}
5062 \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.

```

5063 \newbool{LWR@doingcmidrule}
5064 \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.

```

5065 \newbool{LWR@tableparcell}

```

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

```

5066 \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.

```
5067 \newbool{LWR@skippingmcolrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.

```
5068 \newbool{LWR@skipatbang}
```

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

```
5069 \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.

```
5070 \newbool{LWR@intabularmetadata}
```

```
5071 \boolfalse{LWR@intabularmetadata}
```

61.4 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

```
5072 \newcommand*{\LWR@insertatbangcols}{%
```

```
5073 \ifbool{LWR@skipatbang}%
```

```
5074 {}%
```

```
5075 {%
```

```
5076 \LWR@printatbang{at}{\arabic{LWR@tablecolindex}}%
```

```
5077 \LWR@printatbang{bang}{\arabic{LWR@tablecolindex}}%
```

```
5078 }%
```

```
5079 }
```

\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.

```
5080 \newcommand*{\LWR@closetabledatacell}{%
```

```
5081 \global\booltrue{LWR@intabularmetadata}}%
```

```
5082 \ifbool{LWR@exitingtabular}{}%
```

```
5083 {% not exiting tabular
```

```
5084 \ifboolexpr{bool{LWR@skippingmrowcell} or bool{LWR@skippingmcolrowcell}}%
```

```
5085 {%
```

If not skipping a `\multicolumnrow` cell, insert the @ and ! columns after this non-existent column.

```
5086     \ifbool{LWR@skippingmcolrowcell}%
5087     {}%
5088     {\LWR@insertatbangcols}%
5089     }%
5090     {% not skippingmrowcell
```

Insert any < then any @ and ! column contents, unless muted for the `\bottomrule` or a `\multicolumn`:

```
5091     \unskip%
5092     \ifboolexpr{%
5093         bool{LWR@tabularmutemods} or
5094         bool{LWR@skipatbang} or
5095         bool{LWR@emptyatbang}
5096     }%
5097     {}%
5098     {\LWR@getexparray{LWR@colafterspec}{\arabic{LWR@tablecolindex}}}%
```

Close paragraphs:

```
5099     \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
5100     \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```
5101     \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
5102     \global\booltrue{LWR@tabularcelladded}%
5103     \LWR@insertatbangcols%
5104     }% not skipping mrowcell
5105 }% not exiting tabular
5106 \global\boolfalse{LWR@skippingmrowcell}%
5107 \global\boolfalse{LWR@skippingmcolrowcell}%
5108 \global\boolfalse{LWR@skipatbang}%
5109 }
```

`LWR@tabulardepth` tracks whether & is being used inside a tabular.

```
5110 \newcounter{LWR@tabulardepth}
5111 \setcounter{LWR@tabulardepth}{0}
5112
```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```
5113 \let\LWR@origampmacro&
```

```
5114 \end{warpHTML}
```

61.4.1 Localizing & catcodes

for HTML & PRINT: 5115 \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.

```
5116 \newcommand{\StartDefiningTabulars}{%
5117 \LWR@traceinfo{StartDefiningTabulars}
5118 \warpHTMLonly{\catcode'\&=\active}%
5119 }
```

\EndDefiningTabulars Place after defining something with & in it.

```
5120 \newcommand{\EndDefiningTabulars}{%
5121 \LWR@traceinfo{EndDefiningTabulars}
5122 \warpHTMLonly{\catcode'\&=4}%
5123 }
```

```
5124 \end{warpall}
```

61.4.2 Handling &

for HTML output: 5125 \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.

```
5126 \newcommand*\LWR@tabularampersand{%
5127 \LWR@traceinfo{LWR@tabularampersand}%
5128 \ifnumcomp{\value{LWR@tabulardepth}}{>}{0}%
5129 {%
```

If not skipping a multirow cell, close the current data cell.

```
5130   \unskip%
5131   \LWR@closetabledatacell%
```

Move to the next column.

```
5132   \addtocounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5133   \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
5134   \LWR@getmynexttoken%
5135 }%
```

If not inside a tabular, performs the original action:

```
5136 {\LWR@origampmacro}%
5137 }
```

& 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.

61.4.3 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.

```
5138 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
5139 \ifboolexpr{%
5140   not bool {LWR@exitingtabular} or%
```

```

5141     bool{LWR@doingtbrule} or%
5142     bool{LWR@doingcmidrule} or%
5143     bool{LWR@doinghline} or%
5144     bool{LWR@startedrow}%
5145 }{%

```

To locally temporarily turn off LWR@exitingtabular so that table data tags will still be generated:

```
5146 \begingroup%
```

If generating a final row for the \bottomrule borders, turn off the @, !, <, and > column output:

```

5147 \ifbool{LWR@exitingtabular}{%
5148     \booltrue{LWR@tabularmutemods}%
5149 }{%

```

Reenable the table data tags until finished with the final row:

```
5150 \global\boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```

5151 \whileboolexpr{%
5152     test {
5153         \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
5154     } or %
5155     (%
5156         bool{LWR@intabularmetadata} and%
5157         not bool{LWR@tabularcelladded} and%
5158         test {
5159             \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}
5160         }%
5161     )%
5162 }%
5163 {%
5164     \LWR@tabledatasinglecolumn%

```

The following is essentially \LWR@tabularampersand with LWR@emptyatbang added to empty the following cells:

```

5165     \LWR@closetabledatacell%
5166     \addtocounter{LWR@tablecolindex}{1}%
5167     \boolfalse{LWR@tabularcelladded}%
5168     \global\booltrue{LWR@emptyatbang}%

```

Starts the next cell:

```

5169 \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
5170 {\LWR@getmynexttoken}%
5171 }%
5172 }%

```

Reenable the original LWR@exitingtabular to close the entire table:

```

5173 \endgroup%
5174 \global\boolfalse{LWR@emptyatbang}%
5175 }{}% ifboolexpr
5176 }

```

61.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

\LWR@tabularendofline

```

5177 \NewDocumentCommand{\LWR@tabularendofline}{s o}{%
5178 \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
5179 {\LWR@tabularfinishrow}%
5180 {\LWR@closetabledatacell}%
5181 % \begin{macrocode}
5182 \LWR@htmltag{/tr}\LWR@orignewline
5183 \global\booltrue{LWR@intabularmetadata}

```

Not yet started a table row:

```

5184 \global\boolfalse{LWR@startedrow}

```

Additional setup:

```

5185 \global\boolfalse{LWR@doinghline}%
5186 \global\boolfalse{LWR@doingtbrule}%
5187 \global\boolfalse{LWR@doingcmidrule}%
5188 \LWR@clearmidrules%

```

Start at first column:

```

5189 \setcounter{LWR@tablecolindex}{1}%

```

Have not yet added data in this column:

```

5190 \boolfalse{LWR@tabularcelladded}%

```

Look at the next token to decide between single column data tag or a special case:

```
5191 \LWR@getmynexttoken%
5192 }
```

61.6 Variables

```
5193 \newcommand*\LWR@colsresult{}%temp storage for column format results
5194 \newcommand*\LWR@pposition{}
5195 \newcommand*\LWR@pleft{}
5196 \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.

```
5197 \newcommand*\LWR@tablecolspec{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```
5198 \providecommand*\LWR@strresult{}
5199 \providecommand*\LWR@strresulttwo{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
5200 \newcommand*\LWR@origcolspec{}

```

Ctrl `LWR@tablecolspecwidth` Holds the number of tokens in the table columns specification.

This includes one for each `@`, `!`, `<`, `>` column, and also one for each of the parameters of `p`, `@`, `!`, `<`, `>` columns, and three for each `D` column.

(This is not the total # of \LaTeX columns in the table.)

```
5201 \newcounter{LWR@tablecolspecwidth}

```

Ctrl `LWR@tablecolspecindex` While parsing the \LaTeX table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
5202 \newcounter{LWR@tablecolspecindex}

```

- Ctrl `LWR@tablecolindex` While parsing the \LaTeX table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.
- 5203 `\newcounter{LWR@tablecolindex}`
- Ctrl `LWR@tabletotalcols` While parsing a table column specification, begins at 0 and increments by 1 per \LaTeX table column. Eventually holds the final number of \LaTeX table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in `LWR@tabletotalcols`.)
- 5204 `\newcounter{LWR@tabletotalcols}`
- Ctrl `LWR@tabletotalcolsnext` Holds the next \LaTeX table column index while parsing, equal to one more than `LWR@tabletotalcols`.
- 5205 `\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 35.
- `LWR@colbangspec` A data array of specifications for ! columns. The leftmost's index is `leftedge`, the others are counter values. See section 35.
- `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.

61.7 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

5206 `\newcommand*{\LWR@colparameter}{}`

`\LWR@parseatcolumn` Handles @{text} columns.

5207 `\newcommand*{\LWR@parseatcolumn}{%`

Move to the next token after the '@':

5208 `\LWR@traceinfo{at column}%`

5209 `\addtocounter{LWR@tablecolspecindex}{1}%`

Read the next token into `\LWR@colparameter`, expanding once:

```
5210 \LWR@traceinfo{about to read the next token:}%
5211 \expandarg%
5212 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
5213 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5214 \LWR@traceinfo{have now read the next token}%
5215 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
5216 {% left edge of the table:
5217   \LWR@traceinfo{at the left edge}%
5218   \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
5219   \LWR@traceinfo{at the left edge: %
5220   \LWR@getexparray{LWR@colatspec}{leftedge}}%
5221 }%
5222 {% not at the left edge:
5223   \LWR@traceinfo{not at the left edge}%
5224   \LWR@setexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5225   \LWR@traceinfo{at \arabic{LWR@tabletotalcols}: %
5226   \LWR@getexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}}%
5227 }%
5228 \let\LWR@colparameter\relax%
5229 \booltrue{LWR@validtablecol}%
5230 }
```

`\LWR@parsebangcolumn` Handles `!{text}` columns.

```
5231 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
5232 \LWR@traceinfo{bang column}%
5233 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
5234 \LWR@traceinfo{about to read the next token:}%
5235 \expandarg%
5236 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
5237 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5238 \LWR@traceinfo{have now read the next token}%
5239 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
5240 {% left edge of the table:
```

```

5241 \LWR@traceinfo{at the left edge}%
5242 \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
5243 }%
5244 {% not at the left edge:
5245 \LWR@traceinfo{not at the left edge}%
5246 \LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5247 \LWR@traceinfo{bang \arabic{LWR@tabletotalcols}: \LWR@colparameter!}%
5248 }%
5249 \let\LWR@colparameter\relax%
5250 \booltrue{LWR@validtablecol}%
5251 }

```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```
5252 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
5253 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```

5254 \expandarg%
5255 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5256 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```

5257 \LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tabletotalcolsnext}}{\LWR@colparameter}%
5258 \let\LWR@colparameter\relax%
5259 \booltrue{LWR@validtablecol}%
5260 }

```

`\LWR@parseaftercolumn` Handles `<{text}` columns.

```
5261 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
5262 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```

5263 \expandarg%
5264 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5265 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5266 \LWR@setexparray{\LWR@colafterspec}{\arabic{\LWR@tabletotalcols}}{\LWR@colparameter}%
5267 \let\LWR@colparameter\relax%
5268 \booltrue{\LWR@validtablecol}%
5269 }
```

`\LWR@parsebarcolumn` Handles vertical rules.

```
5270 \newcommand*\LWR@parsebarcolumn{%
5271 \LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
5272 \ifnumcomp{\value{\LWR@tabletotalcols}}{=}{0}%
5273 {% left edge of the table:
5274   \LWR@setexparray{\LWR@colbarspec}{leftedge}{tvertbarl}%
5275 }%
5276 {% not at the left edge:
5277   \LWR@setexparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalcols}}{tvertbarr}%
5278 }%
5279 \booltrue{\LWR@validtablecol}%
5280 }
```

61.8 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.

```
5281 \newcommand*\LWR@parsenormalcolumn}[1]{%
5282 \appto\LWR@tablecolspec{#1}%
5283 \addtocounter{\LWR@tabletotalcols}{1}%
5284 \addtocounter{\LWR@tabletotalcolsnext}{1}%
5285 \LWR@traceinfo{normal column \arabic{\LWR@tabletotalcols}: #1}%
5286 \LWR@setexparray{\LWR@colatspec}{\arabic{\LWR@tabletotalcolsnext}}{%
5287 \LWR@setexparray{\LWR@colbangspec}{\arabic{\LWR@tabletotalcolsnext}}{%
5288 \LWR@setexparray{\LWR@colbeforespec}{\arabic{\LWR@tabletotalcolsnext}}{%
5289 \LWR@setexparray{\LWR@colafterspec}{\arabic{\LWR@tabletotalcolsnext}}{%
5290 \LWR@setexparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalcolsnext}}{%
5291 \booltrue{\LWR@validtablecol}%
5292 }
```

61.9 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{ \langle thiscolumn \rangle \}$ The width will be ignored.

```
5293 \newcommand*\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
5294 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
5295 \addtocounter{LWR@tablecolspecindex}{1}%
5296 }
```

61.10 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{ \langle thiscolumn \rangle \}$ The three parameters will be ignored.

```
5297 \newcommand*\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
5298 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
5299 \addtocounter{LWR@tablecolspecindex}{3}%
5300 }
```

61.11 Parsing the column specifications



HTML CSS cannot exactly match the \TeX concept of a baseline for a table row. Table 8 shows the \TeX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 9 for details.

Table 9 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

Table 8: Tabular baseline

l	p	m	b	r
			bot	
l	par	mid	bot	r
	par	mid		
	par			

Table 9: Tabular HTML column conversions

-
- l, r, c:** Converted to table cells without paragraph tags.
Uses css `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.
- p:** Converted to table cells with paragraph tags. Ref: Table 8, \LaTeX places the top line of a parbox aligned with the rest of the text line, so css `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the css may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
- m:** With paragraph tags, css `vertical-align:middle`.
- b:** With paragraph tags, css `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.
- P, M, B:** Horizontally-centered versions.
- S:** Converted to 'r'. From the `siunitx` package.
- D:** Converted to 'c'. From the `dcolumn` package.
- @, !, >, <:** One each, in that order.
- |:** Vertical rule.
- Unknown:** Converted to 'l'.
- \newcolumn:** Currently treated as unknown.
-

5301 `\newbool{LWR@validtablecol}`

`\LWR@parsetablecols` $\{ \langle \text{colspecs} \rangle \}$

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The final number of cells in each row is stored in `LWR@tabletotalcols`.

5302 `\newcommand*{\LWR@parsetablecols}[1]{%`
 5303 `\LWR@traceinfo{LWR@parsetablecols started}%`

Remember the original supplied column spec:

5304 `\renewcommand*{\LWR@origcolspec}{\#1}%`

Remove spaces:

5305 `\expandarg%`
 5306 `\StrSubstitute{\LWR@origcolspec}{ }{[\LWR@origcolspec]}%`

Clear the parsed resulting column spec:

5307 `\renewcommand*{\LWR@tablecolspec}{\{}}%`

Total number of columns found so far. Also pre-initialize the first several columns of specs:

5308 `\setcounter{LWR@tabletotalcols}{0}%`
 5309 `\setcounter{LWR@tabletotalcolsnext}{1}%`
 5310 `\LWR@setexparray{LWR@colatspec}{leftedge}{\{}}%`
 5311 `\LWR@setexparray{LWR@colatspec}{1}{\{}}%`
 5312 `\LWR@setexparray{LWR@colatspec}{2}{\{}}%`
 5313 `\LWR@setexparray{LWR@colatspec}{3}{\{}}%`
 5314 `\LWR@setexparray{LWR@colbangspec}{leftedge}{\{}}%`
 5315 `\LWR@setexparray{LWR@colbangspec}{1}{\{}}%`
 5316 `\LWR@setexparray{LWR@colbangspec}{2}{\{}}%`
 5317 `\LWR@setexparray{LWR@colbangspec}{3}{\{}}%`
 5318 `\LWR@setexparray{LWR@colbeforespec}{1}{\{}}%`
 5319 `\LWR@setexparray{LWR@colbeforespec}{2}{\{}}%`
 5320 `\LWR@setexparray{LWR@colbeforespec}{3}{\{}}%`
 5321 `\LWR@setexparray{LWR@colafterspec}{1}{\{}}%`
 5322 `\LWR@setexparray{LWR@colafterspec}{2}{\{}}%`
 5323 `\LWR@setexparray{LWR@colafterspec}{3}{\{}}%`
 5324 `\LWR@setexparray{LWR@colbarspec}{leftedge}{\{}}%`
 5325 `\LWR@setexparray{LWR@colbarspec}{1}{\{}}%`
 5326 `\LWR@setexparray{LWR@colbarspec}{2}{\{}}%`
 5327 `\LWR@setexparray{LWR@colbarspec}{3}{\{}}%`

Starting at the first column specification:

```
5328 \setcounter{LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```
5329 \LWR@traceinfo{about to StrLen}%
5330 \expandarg%
5331 \StrLen{\LWR@origcolspec}[\LWR@strresult]%
5332 \fullexpandarg%
5333 \LWR@traceinfo{finished StrLen}%
5334 \setcounter{LWR@tablecolspecwidth}{\LWR@strresult}%
```

Scan through the column specifications:

```
5335 \whileboolexpr{%
5336     not test{%
5337         \ifnumcomp{\value{LWR@tablecolspecindex}}{>}{\value{LWR@tablecolspecwidth}}%
5338     }%
5339 }%
5340 {%
```

Place the next single-character column type into `\LWR@strresult`:

```
5341 \expandarg%
5342 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@strresult]%
5343 \LWR@traceinfo{position \arabic{LWR@tablecolspecindex}: \LWR@strresult}%
5344 \fullexpandarg%
```

Not yet found a valid column type

```
5345 \boolfalse{LWR@validtablecol}%
```

Note that the parameter for a `p{spec}` column is a token list which will NOT match `l,c,r,p`.



```
5346 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
5347 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
5348 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
5349 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{%
5350 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{%
5351 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{%
5352 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{%
5353 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{%
5354 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
5355 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
5356 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
```

```

5357 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
5358 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
5359 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{%
5360 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{%
5361 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{%

```

From the dcolumn package:

```

5362 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%

```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```

5363 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{%

```

Many people define centered versions “P”, “M”, and “B”:

```

\newcolumntype{P}[1]{>\centering\arraybackslash}p{#1}}

```

```

5364 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{%
5365 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{%
5366 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{%

```

If this column was an invalid column type, convert it to an l column:

```

5367 \ifbool{LWR@validtablecol}{}%
5368   \LWR@traceinfo{invalid column type: \LWR@strresult}%
5369   \LWR@parsenormalcolumn{l}%
5370 }%
5371 \addtocounter{LWR@tablecolspecindex}{1}%
5372 }% whiledo
5373 }%

```

61.12 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.

```

5374 \newcommand*{\LWR@maybenewtablerow}
5375 {%
5376 \ifbool{LWR@startedrow}%
5377 {}% started the row
5378 {% not started the row

```

Remember that now have started the row:

```
5379 \global\booltrue{LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
5380 \global\booltrue{LWR@intabularmetadata}%
5381 \ifbool{LWR@doinghline}%
5382 {%
5383   \LWR@htmltag{tr class="hline" }%
5384   \LWR@orignewline%
5385 }%
5386 {% not doing hline
5387   \ifbool{LWR@doingtbrule}%
5388   {%
5389     \LWR@htmltag{tr class="tbrule"}%
5390     \LWR@orignewline%
5391   }%
5392   {\LWR@htmltag{tr}\LWR@orignewline}%
5393 }% end of not doing hline
5394 }% end of not started the row
5395 }
```

61.13 Printing vertical bar tags

```
\LWR@printbartag {<index>}
```

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```
5396 \newcommand*{\LWR@printbartag}[1]{%
5397 \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
5398 {}% muting or empty
5399 {% not muting
5400   \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{#1}}%
5401   \ifdefempty{\LWR@tempone}{\LWR@tempone}%
5402 }% not muting
5403 }
```

61.14 Printing at or bang tags

```
\LWR@printatbang {<at-or-bang>}{<index>}
```

```
5404 \newcommand*{\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```
5405 \edef\LWR@atbangspec{\LWR@getexparray{LWR@col#1spec}{#2}}%
5406 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```
5407 \ifdefempty{\LWR@atbangspec}%
5408 {}%
5409 {% not empty
5410   \LWR@htmltag{%
5411     td class="td#1%
5412     \LWR@subaddcmidruletrim{}{}%
5413     \LWR@printbartag{#2}%
5414     "%
5415     \LWR@tdstartstyles%
5416     \LWR@addcmidrulewidth%
5417     \LWR@tdendstyles%
5418   }%
```

Create an empty cell if muting for the \bottomrule:

```
5419   \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
5420   {}%
5421   {\LWR@atbangspec}%
5422 %
5423   \LWR@htmltag{/td}\LWR@orignewline%
5424   \global\booltrue{LWR@tabularcelladded}%
5425 }% not empty
5426 }%
```

\LWR@addleftmostbartag

```
5427 \newcommand*{\LWR@addleftmostbartag}{%
5428 \ifnumcomp{\value{LWR@tablecolindex}}{=}1}{%
5429   \LWR@printbartag{leftedge}%
5430 }{}%
5431 }
```

\LWR@tabularleftedge

```
5432 \newcommand*{\LWR@tabularleftedge}{%
5433 \ifnumcomp{\value{LWR@tablecolindex}}{=}1}%
5434 {%
5435   \LWR@printatbang{at}{leftedge}%
5436   \LWR@printatbang{bang}{leftedge}%
5437 }% left edge
5438 {}% not left edge
```

5439 }

61.15 Data opening tag

`\LWR@thiscolspec` Temporary storage.

5440 `\newcommand*{\LWR@thiscolspec}{}`

`\LWR@tabledatasinglecolumnstag` Print a table data opening tag with style for alignment

5441 `\newcommand*{\LWR@tabledatasinglecolumnstag}{%`

5442 `{%`

5443 `\LWR@maybenewtablerow%`

Don't start a new paragraph tag if have already started one:

5444 `\ifbool{\LWR@intabularmetadata}{%`

5445 `{%`

If have found the end of tabular command, do not create the next data cell:

5446 `\ifbool{\LWR@exitingtabular}{}%`

5447 `{% not exiting tabular`

Print the @ and ! contents before first column:

5448 `\LWR@tabularleftedge%`

Fetch the current column's alignment character into `\LWR@strresult`:

5449 `\StrChar{\LWR@tablecolspec}{\arabic{\LWR@tablecolindex}}[\LWR@strresult]%`

print the start of a new table data cell:

5450 `\LWR@htmltag{td class="td%`

append this column's spec:

5451 `\LWR@strresult%`

If this column has a `cmidrule`, add "rule" to the end of the HTML class tag. Also add vertical bar tags.

5452 `\LWR@addcmidruletrim%`

```

5453     \LWR@addleftmostbartag%
5454     \LWR@printbartag{\arabic{LWR@tablecolindex}}%
5455     "%

5456     \LWR@tdstartstyles%
5457     \LWR@addcmidrulewidth%
5458     \StrChar{LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[LWR@thiscolspec]%
5459     \LWR@addformatwppalignment{LWR@thiscolspec}%
5460     \LWR@tdendstyles%
5461     }%

```

If this is a p, m, b, or X column, allow paragraphs:

```

5462     \ifboolexpr{%
5463         test{ \ifdefstring{LWR@strresult}{p} } or
5464         test{ \ifdefstring{LWR@strresult}{m} } or
5465         test{ \ifdefstring{LWR@strresult}{b} } or
5466         test{ \ifdefstring{LWR@strresult}{P} } or
5467         test{ \ifdefstring{LWR@strresult}{M} } or
5468         test{ \ifdefstring{LWR@strresult}{B} } or
5469         test{ \ifdefstring{LWR@strresult}{X} }
5470     }%
5471     {% allow pars
5472         \LWR@startpars%
5473         \global\booltrue{LWR@tableparcell}%
5474     }% allow pars
5475     }% no pars

```

Print the > contents unless muted for the \bottomrule:

```

5476     \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
5477     {}%
5478     {%
5479         \LWR@getexparray{LWR@colbefore-spec}{\arabic{LWR@tablecolindex}}%
5480     }%
5481     \global\boolfalse{LWR@intabularmetadata}%
5482     }% not exiting tabular
5483 }{}% in tabular metadata
5484 }%

```

61.16 Midrules

LWR@midrules LWR@midrules is a data array (section 35) 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 35) of columns containing 1 if a midrule

should be left trimmed for each column.

`LWR@trimrrules` `LWR@trimrrules` is a data array (section 35) 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.

```
5485 \newcounter{LWR@midrulecounter}
```

`Len \LWR@heavyrulewidth` The default width of the rule.

```
5486 \newlength{\LWR@heavyrulewidth}
5487 \setlength{\LWR@heavyrulewidth}{.08em}
```

`Len \LWR@lightrulewidth` The default width of the rule.

```
5488 \newlength{\LWR@lightrulewidth}
5489 \setlength{\LWR@lightrulewidth}{.05em}
```

`Len \LWR@cmidrulewidth` The default width of the rule.

```
5490 \newlength{\LWR@cmidrulewidth}
5491 \setlength{\LWR@cmidrulewidth}{.03em}
```

`Len \LWR@thiscmidrulewidth` The width of the next rule, defaulting to `\LWR@cmidrulewidth`.

If not `\LWR@cmidrulewidth`, a style will be used to generate the custom width.

Assigned from the `LWR@midrules` array.

```
5492 \newlength{\LWR@thiscmidrulewidth}
5493 \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.

```
5494 \newcommand*{\LWR@clearmidrules}
5495 {%
5496 \setcounter{LWR@midrulecounter}{1}%
5497 \whileboolexpr{%
5498   not test{%
5499     \ifnumcomp{\value{LWR@midrulecounter}}{>}{\value{LWR@tablecolspecwidth}}%
5500   }%
5501 }%
5502 {%
5503 \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{0pt}%
5504 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
```

```

5505 \LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
5506 \LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
5507 \addtocounter{LWR@midrulecounter}{1}%
5508 }%
5509 }

```

`\LWR@subcmidrule` $\langle width \rangle$ $\langle trim \rangle$ $\langle leftcolumn \rangle$ $\langle rightcolumn \rangle$

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.

```

5510 \newcommand*{\LWR@subcmidrule}[4]{%
5511 \setcounter{LWR@midrulecounter}{#3}%
5512 \whileboolexpr{%
5513   not test {%
5514     \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}%
5515   }%
5516 }%
5517 {%
5518   \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
5519   \addtocounter{LWR@midrulecounter}{1}%
5520 }% whiledo
5521 \IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
5522 \IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
5523 \booltrue{LWR@doingcmidrule}%
5524 }

```

`\LWR@docmidrule` $[\langle width \rangle]$ $\langle trim \rangle$ $\langle leftcolumn-rightcolumn \rangle$

Marks LWR@midrules array elements to be “Y” from left to right columns. Also marks trimming for the L and/or R columns.

```

5525 \NewDocumentCommand{\LWR@docmidrule}{0{\LWR@cmidrulewidth} D(){} >\SplitArgument{1}{-}m}%
5526 {\LWR@subcmidrule{#1}{#2}{#3}}

```

Used to compute margins, tabular trims:

```

5527 \newlength{\LWR@templengthone}%
5528 \newlength{\LWR@templengthtwo}%

```

Used to add a style to a table data cell:

```

5529 \newboolean{LWR@tdhavecellstyle}

```

`\LWR@tdstartstyles` Begins possibly adding a table data cell style.

```
5530 \newcommand*{\LWR@tdstartstyles}{\global\boolfalse\LWR@tdhavecellstyle}}
```

`\LWR@tdaddstyle` Starts adding a table data cell style.

```
5531 \newcommand*{\LWR@tdaddstyle}{%
5532 \ifbool{\LWR@tdhavecellstyle}%
5533 {; }%
5534 { style="%}
5535 \booltrue{\LWR@tdhavecellstyle}%
5536 }
```

`\LWR@tdendstyles` Finishes possibly adding a table data cell style. Prints the closing quote.

```
5537 \newcommand*{\LWR@tdendstyles}{%
5538 \ifbool{\LWR@tdhavecellstyle}{%
5539     "%
5540     \global\boolfalse{\LWR@tdhavecellstyle}%
5541 }{}}%
5542 }
```

`\LWR@subaddcmidruletrim` `{\lefttrim}` `{\righttrim}` Adds a `\cmidrule` with optional trim.

```
5543 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
5544 \setlength{\LWR@templengthone}{\LWR@getexparray{\LWR@midrules}{\arabic{\LWR@tablecolindex}}}%
5545 \ifdimcomp{\LWR@templengthone}{>}{Opt}%
5546 {%
```

Print the class without left and right trim letters appended:

```
5547 \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
5548 \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
5549 }%
5550 {%
5551 \setlength{\LWR@thiscmidrulewidth}{Opt}%
5552 }%
5553 }
```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```
5554 \newcommand*{\LWR@addcmidruletrim}{%
5555 \LWR@subaddcmidruletrim%
```

```

5556 {\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
5557 {\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@tablecolindex}}}%
5558 }

```

`\LWR@addrulewidth` $\langle thiswidth \rangle$ $\langle defaultwidth \rangle$

If not default width, add a custom style with width and color depending on `thiswidth`.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```

5559 \newcommand{\LWR@addrulewidth}[2]{%

```

Only add a custom width if `thiswidth` is different than the `defaultwidth`:

```

5560 \ifboolexpr{%
5561     test{\ifdimcomp{#1}{=}{Opt}}%
5562     or ( test{\ifdimcomp{#1}{=}{#2}} and not bool{FormatWP} )%
5563 }%
5564 {}% default width
5565 {}% custom width

```

Ensure that the width is wide enough to display in the browser:

```

5566     \LWR@forceminwidth{#1}%

```

Begin adding a style:

```

5567     \LWR@tdaddstyle%

```

The style itself:

```

5568     \uselengthunit{PT}%
5569     border-top:\rndprintlength{\LWR@atleastonept} solid %

```

The darkness of the color depends on the thickness of the rule:

```

5570     \ifdimcomp{#1}{<}{\LWR@lightrulewidth}%
5571     {\#AOAOAO}%
5572     {% lightrule or heavier
5573         \ifdimcomp{#1}{<}{\LWR@heavyrulewidth}%
5574         {\#808080}%
5575         {black}%
5576     }% lightrule or heavier
5577 }% custom width
5578 }

```

`\LWR@addcmidrulewidth` Adds a style for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
5579 \newcommand{\LWR@addcmidrulewidth}{%
5580 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5581 }
```

`\LWR@WPcell` $\langle \textit{text-align} \rangle \langle \textit{vertical-align} \rangle$

```
5582 \newcommand*{\LWR@WPcell}[2]{%
5583 \LWR@tdaddstyle%
5584 text-align:#1; vertical-align:#2%
5585 }
```

`\LWR@addformatwpalignment` If `FormatWP`, adds a style for the alignment.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
5586 \newcommand*{\LWR@addformatwpalignment}[1]{%
5587 \ifbool{FormatWP}{%
5588 \IfSubStr{#1}{l}{\LWR@WPcell{left}{middle}}{}%
5589 \IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
5590 \IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%
5591 \IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
5592 \IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
5593 \IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
5594 \IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
5595 \IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
5596 \IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
5597 }{}%
5598 }
```

61.17 Multicolumns

61.17.1 Parsing multicolumns

```
5599 \newcounter{\LWR@tablemulticolwidth}
```

Indexes into the multicolumn specification:

```
5600 \newcounter{\LWR@tablemulticolspos}
```

Remembers multicolumn vertical rules if found in the column spec.

```
5601 \newbool{\LWR@mccolvertbarl}
5602 \newbool{\LWR@mccolvertbarr}
```

`\LWR@printmccoltype` $\langle colspec \rangle$ 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.

```
5603 \newcommand*\LWR@printmccoltype}[1]{%
5604 \LWR@traceinfo{lw@printmccoltype -#1-}%
```

Get one token of the column spec:

```
5605 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
5606 \IfStrEq{\LWR@strresult}{l}{l}{}%
5607 \IfStrEq{\LWR@strresult}{c}{c}{}%
5608 \IfStrEq{\LWR@strresult}{r}{r}{}%
5609 \IfStrEq{\LWR@strresult}{p}{p}{}%
5610 \IfStrEq{\LWR@strresult}{m}{m}{}%
5611 \IfStrEq{\LWR@strresult}{b}{b}{}%
5612 \IfStrEq{\LWR@strresult}{P}{P}{}%
5613 \IfStrEq{\LWR@strresult}{M}{M}{}%
5614 \IfStrEq{\LWR@strresult}{B}{B}{}%
5615 \IfStrEq{\LWR@strresult}{S}{r}{}%
5616 \IfStrEq{\LWR@strresult}{X}{p}{}%

5617 \IfStrEq{\LWR@strresult}{|}{|}{%
5618   \ifnumcomp{value{LWR@tablemulticolspos}}{=}{1}{%
5619     {\booltrue{LWR@mcolvertbarl}}}%
5620     {\booltrue{LWR@mcolvertbarr}}}%
5621 }{}}%
5622 \LWR@traceinfo{lw@printmccoltype done}%
5623 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
5624 \newcommand*\LWR@multicolpartext}{%
5625 \LWR@startpars%
5626 \LWR@multicoltext%
5627 \LWR@stoppars%
5628 }
```

`\LWR@multicolother` $\langle colspec \rangle$ For @, !, >, <, print the next token without paragraph tags:

```
5629 \newcommand*\LWR@multicolother}[1]{%
5630 \addtocounter{LWR@tablemulticolspos}{1}%
5631 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
5632 \LWR@strresult%
```

A valid column data type was found:

```
5633 \booltrue{LWR@validtablecol}%
5634 }
```

\LWR@multicolskip Nothing to print for this column type.

```
5635 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
5636 \booltrue{LWR@validtablecol}%
5637 }
```

\LWR@printmccoldata $\langle\{colspec\}\rangle$ Print the data for any valid column type found.

```
5638 \newcommand*{\LWR@printmccoldata}[1]{%
5639 \LWR@traceinfo{lw@printmccoldata -#1}%
```

Not yet found a valid column type:

```
5640 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
5641 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
5642 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
5643 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
5644 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
5645 \IfStrEq{\LWR@strresult}{D}{}%
5646 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
5647 \LWR@multicoltext%
5648 }{}%
5649 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
5650 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
5651 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
5652 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
5653 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
5654 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
5655 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
5656 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
5657 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
5658 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
```

```

5659 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
5660 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
5661 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%

```

If an invalid column type:

```

5662 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%

```

Tracing:

```

5663 \LWR@traceinfo{lwr@printmccoldata done}%
5664 }

```

```

\parsemulticolumnalignment  {\langle 1: colspec \rangle} {\langle 2: printresults \rangle}

```

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.

```

5665 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
5666 \setcounter{LWR@tablemulticolspos}{1}%
5667 \StrLen{#1}[\LWR@strresult]%
5668 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

5669 \whileboolexpr{%
5670   not test {%
5671     \ifnumcomp{\value{LWR@tablemulticolspos}}{>}{\value{LWR@tablemulticolwidth}}%
5672   }
5673 }%
5674 {%

```

Execute the assigned print function for each token in the column spec:

```

5675 #2{#1}%

```

Move to the next token in the column spec:

```

5676 \addtocounter{LWR@tablemulticolspos}{1}%
5677 }%
5678 }

```

61.17.2 Multicolumn factored code

```
5679 \newcommand{\LWR@multicoltext}{}

```

To find multicolumn right trim:

```
5680 \newcounter{LWR@lastmulticolumn}

```

```
\LWR@domulticolumn  [⟨1: vpos⟩] [⟨2: #rows⟩] {⟨3: numLaTeXcols⟩} {⟨4: numHTMLcols⟩} {⟨5: colspec⟩}
                    {⟨6: text⟩}

```

```
5681 \NewDocumentCommand{\LWR@domulticolumn}{o o m m m +m}{%

```

```
5682 \LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%

```

Remember the text to be inserted, and remember that a valid column type was found:

```
5683 \renewcommand{\LWR@multicoltext}{%

```

```
5684 #6%

```

```
5685 \booltrue{LWR@validtablecol}%

```

```
5686 }%

```

Compute the rightmost column to be included. This is used to create the right trim.

```
5687 \setcounter{LWR@lastmulticolumn}{\value{LWR@tablecolindex}}%

```

```
5688 \addtocounter{LWR@lastmulticolumn}{#3}%

```

```
5689 \addtocounter{LWR@lastmulticolumn}{-1}%

```

Row processing:

```
5690 \LWR@maybe newtable row%

```

Begin the opening table data tag:

```
5691 \LWR@htmltag{td colspan="#4" %

```

```
5692 \IfValueT{#2}{ % rows?

```

```
5693 rowspan="#2" %

```

```
5694 \IfValueT{#1}{% vpos?

```

```
5695 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%

```

```
5696 \ifstrequal{#1}{t}{style="vertical-align:top" }{}%

```

```
5697 }% vpos?

```

```
5698 }% rows?

```

```
5699 class="td%

```

Print the column type and vertical bars:

```
5700 \boolfalse{LWR@mcolvertbarl}%

```

```
5701 \boolfalse{LWR@mcolvertbarr}%
5702 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoltype}%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule” for a horizontal rule:

```
5703 \LWR@subaddcmidruletrim%
5704 {\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
5705 {\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%
```

Also add vertical bar class.

```
5706 \ifbool{LWR@mcolvertbarl}{ tvertbarl}{}%
5707 \ifbool{LWR@mcolvertbarr}{ tvertbarr}{}%
```

Close the class tag’s opening quote:

```
5708 "%
```

```
5709 \LWR@tdstartstyles%
5710 \LWR@addcmidrulewidth%
5711 \LWR@addformatwppalignment{#5}%
5712 \LWR@tdendstyles%
5713 }% end of the opening table data tag
5714 \global\boolfalse{LWR@intabularmetadata}%
5715 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
5716 }
```

61.17.3 Multicolumn

```
\LWR@htmlmulticolumn {<numcols>} {<alignment>} {<text>}
```

```
5717 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
5718 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5719 \LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn tag:

```
5720 \LWR@domulticolumn{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#3}%
```

Move to the next \LaTeX column:

```
5721 \addtocounter{LWR@tablecolindex}{#1}%
5722 \addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
5723 \booltrue{LWR@skipatbang}%
5724 }
```

61.17.4 Longtable captions

longtable captions use \multicolumn.

Bool LWR@starredlongtable Per the caption package, step the counter if longtable*.

```
5725 \newbool{LWR@starredlongtable}
5726 \boolfalse{LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
5727 \providecommand*{\LTcapttype}{table}
```

\LWR@longtabledatacaptiontag * [*toc entry*] {*caption*}

```
5728 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5729 {%
```

Remember the latest name for \nameref:

```
5730 \IfValueTF{#2}{% optional given?
5731   \ifblank{#2}{% optional empty?
5732   {\LWR@setlatestname{#3}}}% empty
5733   {\LWR@setlatestname{#2}}}% given and non-empty
5734 }% optional given
5735 {\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:

```
5736 \LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalcols}}
```

Create the multicolumn tag:

```
5737 \LWR@domulticolumn{\arabic{LWR@tabletotalcols}}{\arabic{LWR@tabhtmlcoltotal}}{P}%
```

```
5738 {% \LWR@domulticolumn
5739 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
5740 {% yes star
5741   \LWR@figcaption%
5742   #3%
5743   \endLWR@figcaption%
5744 }%
5745 {% No star:
```

Not the star version:

Don't step the counter if `\caption[] {A caption.}`

```
5746   \ifbool{LWR@starredlongtable}%
5747   {%
5748     \ifblank{#2}% TOC entry
5749     }%
5750     {%
5751       \refstepcounter{\LTcapttype}%
5752       \protected@edef\@currentlabel{%
5753         \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
5754       }%
5755     }{}%
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
5756   \LWR@figcaption%
5757   \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
5758   \endLWR@figcaption%
```

See if an optional caption was given:

```
5759   \ifblank{#2}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
5760   }{}%
```

If the optional caption was given, but might only be []:

```
5761   {% TOC entry not empty
5762     \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```

5763      {% No TOC entry
5764          \addcontentsline%
5765          {\csuse{ext@LTcapttype}}}%
5766          {\LTcapttype}%
5767          {%
5768          \protect\numberline%
5769          {\csuse{p@LTcapttype}\csuse{the\LTcapttype}}}%
5770          {\ignorespaces #3\protect\relax}%
5771          }%
5772      }% end of No TOC entry

```

The optional caption has text enclosed:

```

5773      {% yes TOC entry
5774          \addcontentsline%
5775          {\csuse{ext@LTcapttype}}}%
5776          {\LTcapttype}%
5777          {%
5778          \protect\numberline%
5779          {\csuse{p@LTcapttype}\csuse{the\LTcapttype}}}%
5780          {\ignorespaces #2\protect\relax}%
5781          }%
5782      }% end of yes TOC entry
5783  }% end of TOC entry not empty
5784 }% end of no star

```

Skip any trailing @ or ! columns for this cell:

```

5785 \booltrue{LWR@skipatbang}%
5786 }% end of \LWR@domulticolumn
5787
5788 \addtocounter{LWR@tablecolindex}{\arabic{LWR@tabletotalcols}}
5789 \addtocounter{LWR@tablecolindex}{-1}
5790
5791 }

```

61.17.5 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.

```
5792 \newcounter{LWR@tabhtmlcolindex}
5793 \newcounter{LWR@tabhtmlcolend}
5794 \newcounter{LWR@tabhtmlcoltotal}
```

`\LWR@subtabularhtmlcolumns` $\langle index \rangle$

Factored from `\LWR@tabularhtmlcolumns`, which follows.

```
5795 \newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the @ specification for this column:

```
5796     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the @ specification is not empty, add to the count:

```
5797     \ifdefempty{\LWR@atbangspec}%
5798         {}%
5799         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
5800     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{#1}}%
5801     \ifdefempty{\LWR@atbangspec}%
5802         {}%
5803         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5804 }
```

`\LWR@tabularhtmlcolumns` $\langle starting \text{\LaTeX} column \rangle$ $\langle number \text{\LaTeX} columns \rangle$

Compute the total number of HTML columns being spanned, considering the starting \LaTeX table column and the number of \LaTeX tabular columns being spanned. Any @ and ! columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```
5805 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the \LaTeX span, to which additional @ and ! columns may be added:

```
5806 \setcounter{LWR@tabhtmlcolindex}{#1}%
5807 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5808 \setcounter{LWR@tabhtmlcolend}{#1}%
5809 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
5810 \ifnumcomp{\value{LWR@tabhtmlcolindex}}{=} {1} {%
5811   \LWR@subtabularhtmlcolumns{leftedge}%
5812 } {%}
```

Walk across the \LaTeX columns looking for @ and ! columns:

```
5813 \whileboolexpr{%
5814   test {%
5815     \ifnumcomp{\value{LWR@tabhtmlcolindex}}{<}{\value{LWR@tabhtmlcolend}}%
5816   }%
5817 }%
5818 {%
5819   \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
5820   \addtocounter{LWR@tabhtmlcolindex}{1}%
5821 }% whiledo
5822 }

5823 \end{warpHTML}
```

61.18 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 192 for the HTML versions.

```
for HTML & PRINT: 5824 \begin{warpall}

\multicolumnrow  {\langle 1:cols \rangle} {\langle 2:halign \rangle} [\langle 3:vpos \rangle] {\langle 4:numrows \rangle} [\langle 5:bigstruts \rangle] {\langle 6:width \rangle} [\langle 7:fixup \rangle]
{\langle 8:text \rangle}
```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:
<https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex>

After the user may have

```
5825 \AtBeginDocument{

  \@ifundefined{@xmultirow} determines if multirow was never loaded.

5826 \@ifundefined{@xmultirow}
5827 {}% no version of multirow was loaded
5828 {}% \@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.

```
5829 \@ifpackageloaded{multirow}{% v2.0 or newer
5830 \ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
5831 {% v2.0+:
5832 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5833   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5834 {\multicolumn{#1}{#2}{\@xmultirow[#3][#4][#5][#6][#7][#8]}}%
5835 }
5836 {% loaded but older, probably not executed:
5837 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5838   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5839 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5][#6][#7][#8]}}%
5840 }
5841 }% 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.

```
5842 {% v1.6 or older did not \ProvidePackage
5843 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5844   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5845 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5][#6][#7][#8]}}%
5846 }
```

The user-level interface. This is provided if the HTML version was not already given.

```
5847 \providecommand*{\multicolumnrow}{\LWR@origmulticolumnrow}
5848 }% \@xmultirow defined, so multirow was loaded
5849 }% AtBeginDocument
```

```
5850 \end{warpall}
```

61.19 Utility macros inside a table

for HTML output: 5851 `\begin{warpHTML}`

```
5852 \newcommand*{\LWR@donothing}{}

```

In case `bigdelim` is not loaded:

```
5853 \newcommand*{\ldelim}{}
5854 \newcommand*{\rdelim}{}

```

```
5855 \end{warpHTML}
```

61.20 Special-case tabular markers

for HTML & PRINT: 5856 \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 8.7.

```
5857 \newcommand*{\TabularMacro}{}

```

```
5858 \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: 5859 \begin{warpHTML}

```

5860 \newcommand*{\ResumeTabular}{%
5861 \global\boolfalse{LWR@exitingtabular}%
5862 \global\boolfalse{LWR@tabularmutemods}%
5863 \LWR@getmynexttoken%
5864 }

```

```
5865 \end{warpHTML}
```

for PRINT output: 5866 \begin{warpprint}

```
5867 \newcommand*{\ResumeTabular}{}

```

```
5868 \end{warpprint}

```

61.21 Checking for a new table cell

for HTML output: 5869 \begin{warpprint}

Bool LWR@exitingtabular When \end is found, turns off the next opening data tag.

```
5870 \newbool{LWR@exitingtabular}

```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```
5871 \newbool{LWR@tabularmutemods}

```

\LWR@tabledatacolumnstag Open a new HTML table cell unless the next token is for a macro which does not create data, such as \hline, \toprule, etc:

```
5872 \newcommand*{\LWR@tabledatacolumnstag}%

```

```
5873 {%

```

```
5874 \LWR@traceinfo{LWR@tabledatacolumnstag}%

```

\show\LWR@mynexttoken to see what tokens to look for

If not any of the below, start a new table cell:

```
5875 \let\mynext\LWR@tabledatasinglecolumnstag%

```

If exiting the tabular:

```
5876 \ifdefequal{\LWR@mynexttoken}{\end}%

```

```
5877   {\global\booltrue{LWR@exitingtabular}}{}%

```

longtable can have a caption in a cell

```
5878 \ifdefequal{\LWR@mynexttoken}{\caption}%

```

```
5879   {\let\mynext\LWR@donothing}{}%

```

Look for other things which would not start a table cell:

```

5880 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
5881     {\let\mynext\LWR@donothing}{}%
5882 \ifdefequal{\LWR@mynexttoken}{\multirow}%
5883     {\let\mynext\LWR@donothing}{}%
5884 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
5885     {\let\mynext\LWR@donothing}{}%

```

If an `\mrowcell`, this is a cell to be skipped over:

```

5886 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
5887     {\let\mynext\LWR@donothing}{}%

```

If an `\mcolrowcell`, this is a cell to be skipped over:

```

5888 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
5889     {\let\mynext\LWR@donothing}{}%
5890 %
5891 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
5892     {\let\mynext\LWR@donothing}{}%
5893 %
5894 \ifdefequal{\LWR@mynexttoken}{\hline}%
5895     {\let\mynext\LWR@donothing}{}%
5896 %
5897 \ifdefequal{\LWR@mynexttoken}{\toprule}%
5898     {\let\mynext\LWR@donothing}{}%
5899 %
5900 \ifdefequal{\LWR@mynexttoken}{\midrule}%
5901     {\let\mynext\LWR@donothing}{}%
5902 %
5903 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
5904     {\let\mynext\LWR@donothing}{}%
5905 %
5906 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
5907     {\let\mynext\LWR@donothing}{}%
5908 %
5909 \ifdefequal{\LWR@mynexttoken}{\cline}%
5910     {\let\mynext\LWR@donothing}{}%
5911 %
5912 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
5913     {\let\mynext\LWR@donothing}{}%
5914 %
5915 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
5916     {\let\mynext\LWR@donothing}{}%
5917 %
5918 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
5919     {\let\mynext\LWR@donothing}{}%
5920 %
5921 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
5922     {\let\mynext\LWR@donothing}{}%

```

```

5923 %
5924 \ifdequal{\LWR@mynexttoken}{\rdelim}%
5925     {\let\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:

```

5926 \mynext%
5927 }

5928 \end{warpHTML}

```

61.22 `\mrowcell`

for HTML & PRINT: 5929 `\begin{warpall}`

`\mrowcell` The user must insert `\mrowcell` into any `\multirow` cells which must be skipped.
 This command has no action during print output.

```

5930 \newcommand*{\mrowcell}{}

5931 \end{warpall}

```

61.23 `\mcolrowcell`

for HTML & PRINT: 5932 `\begin{warpall}`

`\mcolrowcell` The user must insert `\mcolrowcell` into any `\multicolumnrow` cells which must be skipped. This command has no action during print output.


```

5933 \newcommand*{\mcolrowcell}{}

5934 \end{warpall}

```

61.24 New `\tabular` definition

for HTML output: 5935 `\begin{warpHTML}`

These are default definitions in case `booktabs` is not loaded, and are not expected to be used, but must exist as placeholders.

```
5936 \newcommand*\LWR@origtoprule}[1]{\hline}
5937 \newcommand*\LWR@origmidrule}[1]{\hline}
5938 \LetLtxMacro\LWR@origcmidrule\cline
5939 \newcommand*\LWR@origbottomrule}[1]{\hline}
5940 \newcommand*\LWR@origaddlinespace}[1][ ]{}
5941 \newcommand*\LWR@origmorecmidrules{}
5942 \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.

```
5943 \AtBeginDocument{
5944 \ifpackageloaded{lwrap-tables}
5945 {
5946 \newcommand*\LWR@hline}[1][ ]{%
5947   \ifbool{FormatWP}%
5948     {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}%
5949     {\booltrue{LWR@doinghline}}%
5950   \LWR@getmynexttoken}%
5951 }
5952 {
5953 \newcommand*\LWR@hline}{%
5954   \ifbool{FormatWP}%
5955     {\LWR@docmidrule{1-\arabic{LWR@tabletotalcols}}}%
5956     {\booltrue{LWR@doinghline}}%
5957   \LWR@getmynexttoken}%
5958 }
5959 }% 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`.

```
5960 \AtBeginDocument{
5961 \ifundefined{frenchbsetup}%
5962 {% no babel-french
5963   \newcommand*\LWR@nullifyNoAutoSpacing{}
5964 }% no babel-french
5965 {% yes babel-french
5966   \newcommand*\LWR@nullifyNoAutoSpacing}{%
```

```

5967         \NoAutoSpacing%
5968         \renewcommand*\NoAutoSpacing{}%
5969         \renewcommand*\LWR@FBcancel{}%
5970     }
5971 }% yes babel-french
5972 }% AtBeginDocument

```

Env LWR@tabular [*<verticalposition>*] {*<colspecs>*}

The new tabular environment will be \let in \LWR@LwarpStart, since siunitx might redefine tabular in the user's document.

```

5973 \StartDefiningTabulars
5974
5975 \newenvironment*LWR@tabular}[2] []
5976 {%
5977 \LWR@traceinfo{LWR@tabular started}%
5978 \addtocounter{LWR@tabulardepth}{1}%

```

Not yet started a table row:

```
5979 \global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```

5980 \global\boolfalse{LWR@doinghline}%
5981 \global\boolfalse{LWR@doingtbrule}%
5982 \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.

```
5983 \LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the @ and ! columns.

```

5984 \global\boolfalse{LWR@exitingtabular}%
5985 \global\boolfalse{LWR@tabularmutemods}%

```

Create the table tag:

```

5986 \global\booltrue{LWR@intabularmetadata}%
5987 \LWR@forcenewpage
5988 \LWR@htmlblocktag{table}%

```

Parse the table columns:

```
5989 \LWR@parsetablecols{#2}%
```

Table col spec is: `\LWR@tablecolspec` which is a string of `llccrr`, etc.

Do not place the table inside a paragraph:

```
5990 \LWR@stoppars%
```

Track column #:

```
5991 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5992 \boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
5993 \LWR@clearmidrules%
```

`\` becomes a macro to end the table row:

```
5994 \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.

```
5995 \LWR@traceinfo{LWR@tabular: redefining macros}%
5996 \renewcommand*{\hline}{LWR@hline}%
5997 %
5998 \RenewDocumentCommand{\cline}{m}%
5999 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%

6000 \DeclareDocumentCommand{\toprule}{o d()}
6001   {%
6002     \IfValueTF{##1}%
6003       {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6004       {%
6005         \ifbool{FormatWP}%
6006           {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6007           {\booltrue{LWR@doingtbrule}}}%
6008       }%
6009   \LWR@getmynexttoken}%
6010 %
6011 \DeclareDocumentCommand{\midrule}{o d()}%
6012   {%
6013     \IfValueTF{##1}%
```

```

6014         {\LWR@docmidrule[##1]()}{1-\arabic{LWR@tabletotalcols}}}%
6015         {%
6016             \ifbool{FormatWP}%
6017             {\LWR@docmidrule[##1]()}{1-\arabic{LWR@tabletotalcols}}}%
6018             {\booltrue{LWR@doinghline}}}%
6019         }%
6020     \LWR@getmynexttoken}%
6021 %
6022 \DeclareDocumentCommand{\cmidrule}{0{\LWR@cmidrulewidth} d() m}%
6023 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
6024 %
6025 \DeclareDocumentCommand{\bottomrule}{o d()}
6026     {%
6027         \IfValueTF{##1}%
6028             {\LWR@docmidrule[##1]()}{1-\arabic{LWR@tabletotalcols}}}%
6029             {%
6030                 \ifbool{FormatWP}%
6031                 {\LWR@docmidrule[##1]()}{1-\arabic{LWR@tabletotalcols}}}%
6032                 {\booltrue{LWR@doingtbrule}}}%
6033             }%
6034     \LWR@getmynexttoken}%
6035 %
6036 \DeclareDocumentCommand{\addlinespace}{o}{}%
6037 \DeclareDocumentCommand{\morecmidrules}{-}{-}%
6038 \DeclareDocumentCommand{\specialrule}{m m m d()}%
6039     {\LWR@docmidrule[##1]()}{1-\arabic{LWR@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.

```

6040 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
6041 \renewcommand*\mrowcell{%
6042     \LWR@maybenewtablerow%
6043     \LWR@tabularleftedge%
6044     \global\booltrue{LWR@skippingmrowcell}%
6045 }%
6046 \renewcommand*\mcolrowcell{%
6047     \LWR@maybenewtablerow%
6048     \global\booltrue{LWR@skippingmcolrowcell}%
6049 }%
6050 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%

```

Reset for new processing:

```

6051 \global\boolfalse{LWR@tableparcell}%
6052 \global\boolfalse{LWR@skippingmrowcell}%
6053 \global\boolfalse{LWR@skippingmcolrowcell}%
6054 \global\boolfalse{LWR@skipatbang}%
6055 \global\boolfalse{LWR@emptyatbang}%

```

Set & for its special meaning inside the tabular:

```
6056 \StartDefiningTabulars%
6057 \protected\gdef&{\LWR@tabularampersand}%
```

Look ahead for a possible table data cell:

```
6058 \LWR@traceinfo{\LWR@tabular: about to LWR@getmynexttoken}%
6059 \LWR@getmynexttoken%
6060 }%
```

Ending the environment:

```
6061 {%
6062 \LWR@traceinfo{\LWR@tabular ending}%
6063 \ifboolexpr{%
6064     test {%
6065         \ifnumcomp{\value{\LWR@tablecolindex}}{<}{\value{\LWR@tabletotalcols}}
6066     } or %
6067     (%
6068         bool{\LWR@intabularmetadata} and%
6069         not bool{\LWR@tabularcelladded} and%
6070         test {%
6071             \ifnumcomp{\value{\LWR@tablecolindex}}{=}{\value{\LWR@tabletotalcols}}%
6072         }%
6073     )%
6074 }%
6075 {%
6076     \LWR@tabularfinishrow%
6077 }%
6078 {%
6079     \LWR@closetabledatacell%
6080 }%
6081 \LWR@htmlblocktag{/tr}%
6082 \LWR@htmlblocktag{/table}%
6083 \global\boolfalse{\LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
6084 \addtocounter{\LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
6085 \protected\gdef&{\LWR@origampmacro}%
6086 \EndDefiningTabulars%
6087 \LWR@traceinfo{\LWR@tabular finished ending}%
6088 }
6089
6090 \EndDefiningTabulars
```

```
6091 \end{warpHTML}
```

61.25 Array

Pkg array

array is also automatically loaded by siunitx.

62 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The zref package is used to remember section name, file, and lateximage depth and number for each label.

Table 10 shows the data structures related to cross-referencing.

for HTML output:

```
6092 \begin{warpHTML}
```

62.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
6093 \providecommand*\@currentlabelname{-}
```

`\LWR@stripperperiod` `{\text}` [`<.>`]

Removes a trailing period.

```
6094 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` `{\object name}`

Removes `\label`, strips any final period, and remembers the result.

```
6095 \newcommand*\LWR@setlatestname[1]{%
```

Table 10: Cross-referencing data structures

Original \LaTeX:	(print and HTML)
<p><code>\refstepcounter</code>: Steps the counter and sets <code>\@currentlabel</code>.</p> <p><code>\@currentlabel</code>: <code>\p@<ctr>\the<ctr></code> Updated by <code>\refstepcounter</code>.</p> <p><code>\label</code>: Writes to the .aux file: <code>\newlabel{<label>}{\@currentlabel}{\thepage}}</code></p> <p><code>\newlabel</code>: When the .aux file is read, sets <code>\r@<label></code>.</p> <p><code>\r@<label></code>: Set to: <code>{\@currentlabel}{\thepage}}</code></p> <p><code>\ref</code>: Returns the first part of <code>\r@<label></code>.</p> <p><code>\pageref</code>: Returns the second part of <code>\r@<label></code>.</p>	
Added by lwarp:	(HTML only)
<p><code>\label</code>: Adds HTML tags (section 62.3), plus <code>\splabel</code> data (section 62.2):</p> <p style="padding-left: 2em;">zLWR@name: The section name for this label.</p> <p style="padding-left: 2em;">zLWR@htmlfilenumber: The file number or name for this label.</p> <p style="padding-left: 2em;">zLWR@lateximagedepth: The <code>lateximagedepth</code> for this label.</p> <p style="padding-left: 2em;">zLWR@lateximagenumber: The <code>lateximagenumber</code> for this label.</p> <p><code>\nameref</code>: Emulated from <code>hyperref</code> for <code>lwarp</code>. See section 62.4.</p> <p><code>\ref</code> and <code>\nameref</code>: Adds HTML tags. See section 62.4.</p>	
Added by amsmath:	(print and HTML)
<p><code>\label</code>: Execution is delayed until the math environment is completed.</p> <p><code>\ltx@label</code>: \LaTeX <code>\label</code>, (HTML: patched by <code>lwarp</code>), later patched by <code>cleveref</code>.</p>	
Added by cleveref:	(print and HTML)
<p><code>\refstepcounter</code>: Added: sets <code>\cref@currentlabel</code>.</p> <p><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 48.4 for use with footnotes.</p> <p><code>\label</code>: Writes to the .aux file: <code>\newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}</code></p> <p><code>\newlabel</code>: (Unchanged.) When the .aux file is read, sets <code>\r@<label>@cref</code>.</p> <p><code>\r@<label>@cref</code>: Set to: <code>{\cref@currentlabel}{\thepage}}</code></p> <p>Utility functions: See <code>\cref@getlabel</code>, <code>\cref@gettype</code>, <code>\cref@getcounter</code>, <code>\cref@getprefix</code>.</p> <p>Cross-referencing names: <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.</p>	
Additionally patched by lwarp:	(HTML only)
<p><code>\cref</code>, etc.: Modified for <code>lwarp</code>. See section 74.</p> <p><code>\label</code> inside math: See section 67.5.1.</p>	
Footnotes : See <code>\noteentry</code> in section 48.4.	

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `getttitlestring`.

```
6096 \GetTitleStringExpand{#1}%
6097 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
6098 \edef\@currentlabelname{%
6099 \expandafter\LWR@stripperperiod\@currentlabelname%
6100 \ltx@empty.\ltx@empty\@nil%
6101 }%
6102 }
```

62.2 Zref setup

See:

<http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference>

Create a new property list called `special`:

```
6103 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
6104 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
6105 \zref@newprop{zLWR@htmlfilenumber}{%
6106 \ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}%
6107 }%
```

Additional properties for `lateximages`:

```
6108 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
6109 \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`:

```
6110 \zref@addprop{special}{zLWR@name}
6111 \zref@addprop{special}{zLWR@htmlfilenumber}
6112 \zref@addprop{special}{zLWR@lateximagedepth}
6113 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
6114 \newcommand*\LWR@sprof}[2]{%
6115 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` $\langle label \rangle$ Returns the section name for this label:

```
6116 \newcommand*\LWR@nameref}[1]{%
6117 \LWR@sprof{#1}{zLWR@name}%
6118 }
```

`\LWR@htmlfileref` $\langle label \rangle$ Returns the file number for this label:

```
6119 \newcommand*\LWR@htmlfileref}[1]{%
6120 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
6121 \LWR@sprof{#1}{zLWR@htmlfilenumber}%
6122 }
```

`\LWR@lateximagedepthref` $\langle label \rangle$ Returns the lateximagedepth for this label:

```
6123 \newcommand*\LWR@lateximagedepthref}[1]{%
6124 \LWR@sprof{#1}{zLWR@lateximagedepth}%
6125 }
```

`\LWR@lateximagenumberref` $\langle label \rangle$ Returns the lateximagenumber for this label:

```
6126 \newcommand*\LWR@lateximagenumberref}[1]{%
6127 \LWR@sprof{#1}{zLWR@lateximagenumber}%
6128 }
```

`\LWR@splabel` $\langle label \rangle$ Sanitize the name and then creates the label:

```
6129 \newcommand*\LWR@splabel}[1]{%
6130 \LWR@setlatestname{\@currentlabelname}%
6131 \zref@labelbylist{#1}{special}}
```

62.3 Labels

`\LWR@subsublabel` $\langle label \rangle$ Creates an HTML id tag.

```
6132 \newcommand*\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
6133 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
6134 {}%
6135 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
6136   \ifbool{LWR@doingstartpars}%
6137   {% pars allowed
6138     \ifbool{LWR@doingapar}
6139     {% par started
6140       \LWR@htmltag{a id="#1"}\LWR@htmltag{/a}%
6141     }% par started
6142   {% par not started
6143     \LWR@stoppars%
6144     \LWR@htmltag{a id="#1"}\LWR@htmltag{/a}%
6145     \LWR@startpars%
6146   }% par not started
6147 }% pars allowed
6148 {% pars not allowed
6149   \LWR@htmltag{a id="#1"}\LWR@htmltag{/a}%
6150 }% pars not allowed
6151 }% not lateximage
6152 }
```

`\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.

```
6153 \NewDocumentCommand{\LWR@newlabel}{m o}{%
6154 \LWR@traceinfo{LWR@newlabel: starting}%
6155 \LWR@traceinfo{LWR@newlabel: !#1!}%
6156 % \@bsphack%
```

Create a traditional \LaTeX label, as modified by `cleveref`:

```
6157 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

6158 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
6159 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
6160 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
6161 \LWR@splabel{#1}%
6162 \LWR@subsublabel{#1}%
6163 % \@esphack%
6164 \LWR@traceinfo{LWR@newlabel: done}%
6165 }

```

62.4 References

`\LWR@startref` $\{\langle label \rangle\}$ (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```

6166 \newcommand*{\LWR@startref}[1]
6167 {%
6168 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
6169 \LWR@traceinfo{LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

6170 \LWR@htmltag{a href="%
6171 \LWR@traceinfo{LWR@startref B}%
6172 \LWR@htmlrefsectionfilename{#1}}%
6173 \LWR@traceinfo{LWR@startref C}%
6174 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

6175 \LWR@traceinfo{LWR@startref D: !#1!}%
6176 \ifthenelse{\equal{\LWR@lidref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

6177 {%
6178 \LWR@traceinfo{LWR@startref D0: ??}%
6179 ??%
6180 }%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

6181 {%
6182   \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
6183   \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
6184   {%
6185     \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
6186     lateximage\LWR@lateximagenumberref{#1}%
6187   }%
6188   {%
6189     \LWR@traceinfo{LWR@startref D3}%
6190     #1%
6191   }%
6192 }%
6193 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

6194 "}%
6195 \LWR@traceinfo{LWR@startref F}%
6196 }

```

`\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)”.

```

6197 \NewDocumentCommand{\LWR@subnewref}{m m}{%
6198 \LWR@traceinfo{LWR@subnewref #1 #2}%
6199 \LWR@startref{#1}%
6200 \LWR@origref{#2}%
6201 \LWR@htmltag{/a}%
6202 }

```

`\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.

```

6203 \NewDocumentCommand{\LWR@newref}{s m}{%
6204 \LWR@traceinfo{LWR@newref #2}%
6205 \IfBooleanTF{#1}%
6206 {\LWR@origref{#2}}%
6207 {\LWR@subnewref{#2}{#2}}%
6208 }

```

`\pagerefPageFor` Text for page references.

```

6209 \newcommand*{\pagerefPageFor}{see }

```

`\pageref` * $\langle label \rangle$ Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```
6210 \NewDocumentCommand{\LWR@newpageref}{s m}{%
6211 \IfBooleanTF{#1}%
6212 {(\pagerefPageFor\LWR@origref{#2})}%
6213 {(\cpageref{#2})}%
6214 }
```

`\nameref` $\langle label \rangle$

```
6215 \DeclareRobustCommand*\nameref[1]{%
6216 \LWR@traceinfo{nameref A}%
6217 \LWR@startref{#1}%
6218 \LWR@traceinfo{nameref B}%
6219 \LWR@nameref{#1}%
6220 \LWR@traceinfo{nameref C}%
6221 \LWR@htmltag{/a}%
6222 \LWR@traceinfo{nameref D}%
6223 }
```

`\Nameref` $\langle label \rangle$ In print, adds the page number. In HTML, does not.

```
6224 \let\Nameref\nameref
```

62.5 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.

```
6225 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
6226 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref` $\langle URL \rangle$ $\langle text \rangle$

```
6227 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
6228 \LWR@htmltag{a href="#1" target="_{blank}"\LWR@orignewline}#2\LWR@htmltag{/a}%
6229 \LWR@ensuredoingapar%
6230 }
```

`\LWR@subhyperrefclass` $\langle URL \rangle$ $\langle text \rangle$ $\langle htmlclass \rangle$

```
6231 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
6232 \LWR@htmltag{a href="#1"
6233 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
6234 \LWR@ensuredoingapar%
6235 }
```

`\href` [$\langle options \rangle$] $\langle URL \rangle$ $\langle text \rangle$

Create a link with accompanying text:

```
6236 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
6237 \LWR@subhyperref{#2}{#3}%
6238 \endgroup%
6239 \LWR@ensuredoingapar%
6240 }
6241
6242 \DeclareRobustCommand*\href}{%
6243 \LWR@ensuredoingapar%
6244 \begingroup%
6245 \catcode'\_ =12
6246 \LWR@hrefb%
6247 }
```

`\nolinkurl` $\langle URL \rangle$

Print the name of the link without creating the link:

```
6248 \newcommand*\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
6249
6250 \DeclareRobustCommand*\nolinkurl}{%
6251 \LWR@ensuredoingapar%
6252 \begingroup\catcode'\_ =12
6253 \LWR@nolinkurlb%
6254 }
```

`\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`.

```
6255 \NewDocumentCommand{\LWR@urlb}{m}{%
6256 \LWR@hrefb{#1}{#1}%
6257 }
6258
6259 \DeclareRobustCommand*{\url}{%
6260 \LWR@ensuredoingapar%
6261 \begingroup\catcode'\_ =12
6262 \LWR@urlb%
6263 }
6264
6265 % \LetLtxMacro\url\LWR@urla
```

`\LWR@subinlineimage` [`\alttag`] [`\class`] [`\filename`] [`\extension`] [`\style`]

```
6266 \newcommand*{\LWR@subinlineimage}[5] [] {%
6267 \ifblank{#1}%
6268 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{} }}%
6269 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{} }}%
6270 }

6271 \end{warpHTML}
```

Table 11: Float data structures

For each `<type>` of float (figure, table, etc.) there exists the following:

counter <type>: A counter called `<type>`, such as figure, table.

`\<type>name`: Name. `\figurename` prints “Figure”, etc.

`\ext@<type>`: File extension. `\ext@figure` prints “lof”, etc.

`\fps@<type>`: Placement.

`\the<type>`: Number. `\thetable` prints the number of the table, etc.

`\p@<type>`: Parent’s number. Prints the number of the [within] figure, etc.

`\fnum@<type>`: Prints the figure number for the caption.

`\<type>name \the<type>`, “Figure 123”.

`\<type>`: Starts the float environment. `\figure` or `\begin{figure}`

`\end<type>`: Ends the float environment. `\endfigure` or `\end{figure}`

`\tf@<ext>`: The \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.

63 Floats

Floats are supported, although partially through emulation.

Table 11 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

63.1 Float captions

for HTML output: 6272 `\begin{warpHTML}`

`\LWR@floatbegin` $\langle type \rangle$ [$\langle placement \rangle$]

Begins a `\newfloat` environment.

```
6273 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
6274 \ifbool{FormatWP}{\newline}{}%
6275 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
6276 \addtocounter{LWR@thisautoid}{1}%
6277 \booltrue{LWR@freezethisautoid}%
```

```
6278 \begingroup%
```

Settings while inside the environment:

```
6279 \LWR@origraggedright%
```

Open an HTML figure tag:

```
6280 \LWR@htmltag{figure id="autoid-\arabic{LWR@thisautoid}" class="#1"}%
6281 \ifbool{FormatWP}{%
6282   \LWR@orignewline%
6283   \LWR@BlockClassWP}{-}{wp#1}%
6284 }{}%
```

```
6285 \renewcommand*{\@capttype}{#1}%
6286 \caption@settype{#1}%
6287 \LWR@startpars%
6288 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6289
6290 === begin #1 ===
6291
6292 }{}%
6293 }
```

`\@float` Support packages which create floats directly.
`\@dblfloat`

```
6294 \let\@float\LWR@floatbegin
6295 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
6296 \newcommand*{\LWR@floatend}{%
6297 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6298
6299 === end ===
6300
6301 }{}}%
6302 \LWR@stoppars%
```

Close an HTML figure tag:

```
6303 \ifbool{FormatWP}{\endLWR@BlockClassWP}{}}%
6304 \LWR@htmllementend{figure}%
6305 \endgroup%
6306 \boolfalse{LWR@freezethisautoid}%
6307 \LWR@startpars%
6308 \ifbool{FormatWP}{\newline}{}}%
6309 }
```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```
6310 \let\end@float\LWR@floatend
6311 \let\end@dblfloat\LWR@floatend
```

Ctrl `LWR@thisautoid` A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

```
6312 \newcounter{LWR@thisautoid}
```

Bool `LWR@freezethisautoid` Prevents multiple increments of `\LWR@thisautoid` inside a float.

```
6313 \newbool{LWR@freezethisautoid}
6314 \boolfalse{LWR@freezethisautoid}
```

`\LWR@maybeinthisautoid` Possibly increments the autoid counter.

```
6315 \newcommand*{\LWR@maybeinthisautoid}{%
6316 \ifbool{LWR@freezethisautoid}{\addtocounter{LWR@thisautoid}{1}}%
6317 }
```

`\LWR@newautoidanchor` Adds a new `<autoid>` anchor.

```
6318 \newcommand*{\LWR@newautoidanchor}{%
6319 \LWR@maybeinthisautoid%
6320 \LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}"}{\LWR@htmltag{/a}%
6321 }
```

`\@capttype` Remembers which float type is in use.

```
6322 \newcommand*{\@capttype}{}
```

63.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```
6323 \AtBeginDocument{\providecommand*\CaptionSeparator}{:-~}}
```

`\@makecaption` `{\langle name and num \rangle}{\langle text \rangle}`

Prints the float type and number, the caption separator, and the caption text.

```
6324 \AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}
```

63.1.2 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 `autopage HTML <id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an `HTML autoid <id>` at the start of the float itself in the HTML file. The `autopage` and `autoid` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctrl `LWR@nextautoid` Tracks autoid for floats. Tracks autopage for floats.

Ctrl `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
6325 \newcounter{LWR@nextautoid}
```

```
6326 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat` `{\langle autopage \rangle}{\langle autoid \rangle}`

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autoid` are remembered for `\l@figure` to use when creating the HTML links.

```
6327 \newcommand*\LWRsetnextfloat}[2]{%
```

```
6328 \setcounter{LWR@nextautopage}{#1}%
```

```
6329 \setcounter{LWR@nextautoid}{#2}%
```

```
6330 }
```

Ctrl LWR@latestautopage Updated each time a new HTML file is begun. \LWRsetnextfloat is written with this and the autoid by the modified \addcontentsline just before each float's entry.

```
6331 \newcounter{LWR@latestautopage}
6332 \setcounter{LWR@latestautopage}{1}
```

Env LWR@figcaption Encapsulates a caption inside <figcaption>, and if FormatWP then also a <div> with an italic style.

```
6333 \newenvironment*{LWR@figcaption}
6334 {%
6335 \LWR@htmlblocktag{figcaption}
6336 \ifbool{FormatWP}{%
6337 \begin{BlockClass}[font-style:italic]{italic}
6338 \LWR@origvspace*{\baselineskip}
6339 }{}%
6340 }
6341 {%
6342 \ifbool{FormatWP}{\end{BlockClass}}{}%
6343 \LWR@htmlblocktag{/figcaption}%
6344 }
```

```
6345 \let\LWR@origcaption@begin\caption@begin
6346 \let\LWR@origcaption@end\caption@end
6347 \let\LWR@orig@@par\@@par
```

\LWR@caption@begin Low-level patches to create HTML tags for captions.

```
6348 \newcommand{\LWR@caption@begin}
6349 {
6350 \LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
6351 \begingroup%
```

The caption code was not allowing the closing par tag:

```
6352 \renewcommand{\@@par}{\LWR@closeparagraph\LWR@orig@@par}%
```

No need for a minipage or \parbox inside the caption:

```
6353 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
6354 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```

6355 \LWR@figcaption%
6356 \LWR@origcaption@begin%
6357 }

```

\LWR@caption@end Low-level patches to create HTML tags for captions.

```

6358 \newcommand{\LWR@caption@end}
6359 {%
6360 \LWR@origcaption@end%

```

Subcaptions were being over-written by the closing HTML tag:

```

6361 \LWR@origvspace*{\baselineskip}%

```

Closing tag:

```

6362 \endLWR@figcaption%
6363 \endgroup%
6364 % \leavevmode% avoid bad space factor (0) error
6365 \LWR@traceinfo{\LWR@caption@end: done}%
6366 }

```

\caption@begin Low-level patches to create HTML tags for captions.
 \caption@end

```

6367 \AtBeginDocument{
6368 \let\caption@begin\LWR@caption@begin
6369 \let\caption@end\LWR@caption@end
6370 }

```

\captionlistentry Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

6371 \let\LWR@origcaptionlistentry\captionlistentry
6372
6373 \renewcommand*{\captionlistentry}{%
6374 \LWR@ensuredoingapar%
6375 \LWR@origcaptionlistentry%
6376 }
6377
6378 \def\LWR@LTcaptionlistentry{%
6379 \LWR@ensuredoingapar%
6380 \LWR@htmltag{a id="autoid-\arabic{\LWR@thisautoid}"}{\LWR@htmltag{/a}%
6381 \bgroup
6382 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
6383 {\egroup\LWR@LTcaptionlistentry}}%
6384 \def\LWR@LTcaptionlistentry#1{%
6385 \caption@listentry\@firstoftwo[LTcapye]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autoid before each float's entry. No changes if writing `.toc`. For a theorem, automatically defines `\ext@<type>` as needed, to mimic and reuse the float mechanism.

```

6386 \let\LWR@origaddcontentsline\addcontentsline
6387
6388 \renewcommand*\addcontentsline}[3]{%
6389 \ifstrequal{#1}{toc}{-}{% not TOC
6390   \LWR@newautoidanchor%
6391   \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{-}%
6392   \addtocontents{\@nameuse{ext@#2}}{%
6393     \protect\LWRsetnextfloat%
6394     {\arabic{LWR@latestautopage}}%
6395     {\arabic{LWR@thisautoid}}%
6396   }%
6397 }% not TOC
6398 \LWR@origaddcontentsline{#1}{#2}{#3}%
6399 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

6400 \AtBeginDocument{
6401 \let\LWR@origcaptionof\captionof
6402
6403 \renewcommand*\captionof{%
6404 \LWR@stoppars
6405 \LWR@origcaptionof%
6406 }
6407 }

6408 \end{warpHTML}

```

64 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: 6409 `\begin{warpHTML}`

64.1 Reading and printing the TOC

`\LWR@myshorttoc` `{\{toc/lof/lot}}`

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 sideroc.

```
6410 \newcommand*{\LWR@myshorttoc}[1]{
6411 \LWR@ensuredoingapar
```

Only if the file exists:

```
6412 \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.



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`.

```
6413 \begingroup
6414 % \ifxetexorluatex%
6415 % \else
6416 % \inputencoding{latin1}% currently disabled
6417 % \fi
6418 \makeatletter
```

Read in the TOC file:

```
6419 \@input{\jobname.#1}
6420 % \makeatother
6421 \endgroup
6422 }%
6423 {}%
6424 }
```

`\LWR@subtableofcontents` `{\{toc/lof/lot}} {\{sectionstarname}}`

Places a TOC/LOF/LOT at the current position.

```
6425 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
6426 \@ifundefined{chapter}
6427 {\LWR@closeprevious{\LWR@depthsection}}
6428 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
6429 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
6430 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
6431 \LWR@htmlclass{nav}{#1}
```

Create the actual list:

```
6432 \LWR@myshorttoc{#1}
```

Close the nav:

```
6433 \LWR@htmlclassend{nav}{#1}
6434 }
```

```
\starttoc {<ext>}
```

Patch `\starttoc` to encapsulate the TOC inside HTML tags:

```
6435 \let\LWR@orig@starttoc\starttoc
6436
6437 \renewcommand{\starttoc}[1]{
6438 \LWR@htmlclass{nav}{#1}
6439 \LWR@orig@starttoc{#1}
6440 \LWR@htmlclassend{nav}{#1}
6441 }
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```
6442 \let\LWR@origtableofcontents\tableofcontents
```

```
6443
6444 \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:

```
6445 \ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{
6446
6447 === table of contents ===
6448
6449 }
6450 {
```

Copy the .toc file to .sidetoc for printing the sideroc. The original .toc file is renewed when \tableofcontents is finished.

```
6451     \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
6452     \LWR@printpendingfootnotes
6453     \LWR@origtableofcontents
6454 }
6455 }
```

\listoffigures

```
6456 \let\LWR@origlistoffigures\listoffigures
6457
6458 \renewcommand*{\listoffigures}{
6459 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
6460
6461 === list of figures ===
6462
6463 }
6464 {
6465     \LWR@printpendingfootnotes
6466     \LWR@origlistoffigures
6467 }
6468 }
```

\listoftables

```
6469 \let\LWR@origlistoftables\listoftables
6470
6471 \renewcommand*{\listoftables}{
6472 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
6473
6474 === list of tables ===
6475
6476 }
```

```

6477 {
6478   \LWR@printpendingfootnotes
6479   \LWR@origlistoftables
6480 }
6481 }

```

64.2 High-level TOC commands

`\listof` $\langle type \rangle$ $\langle title \rangle$

Emulate the `\listof` command from the float package (section 141). Used to create lists of custom float types. Also used to redefine the standard \LaTeX `\listoffigures` and `\listoftables` commands.

```

6482 \NewDocumentCommand{\listof}{m +m}{%
6483 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
6484 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
6485 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
6486   \jobname.\csuse{ext@#1}\relax
6487 }

```

64.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

css may be used to format the sideTOC:

CSS related to sideTOC:

nav.sidetoc: The entire sidetoc.

div.sidetoctitle: The title.

div.sidetoccontents: The table of contents.

```

6488 \end{warpHTML}

```

for HTML & PRINT: 6489 `\begin{warpall}`

Ctrl `SideTOCDepth` Controls how deep the side-TOC gets. Use a standard \LaTeX section level similar to

tocdepth.

```
6490 \newcounter{SideTOCDepth}
6491 \setcounter{SideTOCDepth}{1}
```

`\sitetocname` Holds the default name for the sidetoc.

```
6492 \newcommand{\sitetocname}{Contents}
```

```
6493 \end{warpall}
```

for HTML output: 6494 `\begin{warpHTML}`

`\LWR@sitetoc` Creates the actual side-TOC.

```
6495 \newcommand*{\LWR@sitetoc}{
6496 \LWR@forcenewpage
6497 \LWR@stoppars
6498
```

The entire sidetoc is placed into a nav of class sidetoc.

```
6499 \LWR@htmlclass{nav}{sitetoc}
6500
6501 \setcounter{tocdepth}{\value{SideTOCDepth}}
6502
```

The title is placed into a `<div>` of class sidetoc title, and may contain paragraphs.

```
6503 \begin{BlockClass}{sitetoc title}
6504 \sitetocname
6505 \end{BlockClass}
```

The table of contents is placed into a `<div>` of class sidetoc contents.

```
6506 \begin{BlockClass}{sitetoc contents}
6507 \LinkHome
6508
6509 \LWR@myshorttoc{sitetoc}
6510 \end{BlockClass}
6511 \LWR@htmlclassend{nav}{sitetoc}
6512 }
```

64.4 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:

```
6513 \newcommand*\LWR@numberline}[1]{%
6514 \LWR@sectionnumber{#1}\quad%
6515 }
6516
6517 \LetLtxMacro\numberline\LWR@numberline
```

`\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

```
6518 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
6519 \ifthenelse{\cinttest{#1}{<=} {\value{tocdepth}}}{%
6520 \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
6521 \LWR@subhyperrefclass{%
6522 \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
6523 \LWR@stoppars%
6524 }
6525 {}
6526 }
```

Ctrl `lofdepth` TOC depth for figures.

```
6527 \newcounter{lofdepth}
6528 \setcounter{lofdepth}{1}
```

Ctrl `lotdepth` TOC depth for tables.

```
6529 \newcounter{lotdepth}
6530 \setcounter{lotdepth}{1}
```

`\hypertocfloat` $\langle 1: depth \rangle \langle 2: type \rangle \langle 3: ext\ of\ parent \rangle \langle 4: caption \rangle \langle 5: page \rangle$

#1 is depth

#2 is figure, table, etc.

#3 is lof, lot, of the parent.

#4 the text of the caption

#5 page number

```
6531 \newcommand{\hypertocfloat}[5]{%
6532 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
6533 \@ifundefined{c#3depth}{%
6534 \newcounter{#3depth}%
6535 \setcounter{#3depth}{1}%
6536 }{}%
```

Respond to `lofdepth`, etc.:

```
6537 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
6538 \ifthenelse{\cnttest{#1}{<=} {\arabic{#3depth}}}{%
6539 \LWR@startpars%
```

Create an HTML link to `filename#autoid-(float number)`, with text of the caption, of the given HTML class.

```
6540 \LWR@subhyperrefclass{%
6541 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
6542 \#autoid-\arabic{LWR@nextautoid}}%
6543 {#4}{toc#2}%
6544 \LWR@stoppars%
6545 }{}%
6546 }
```

Automatically called by `\contentsline`:

```

6547 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
6548 \DeclareDocumentCommand{\l@chapter}{m m}
6549     {\hypertoc{0}{chapter}{#1}{#2}}
6550 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
6551 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
6552 \renewcommand{\l@subsubsection}[2]
6553     {\hypertoc{3}{subsubsection}{#1}{#2}}
6554 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
6555 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
6556 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
6557 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

6558 \end{warpHTML}

```

65 Index and glossary

See:

[http://tex.stackexchange.com/questions/187038/
how-to-mention-section-number-in-index-created-by-imakeidx](http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx)

Index links are tracked by the counter `LWR@autoindex`. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output: 6559 `\begin{warpHTML}`

```

6560 \newcounter{LWR@autoindex}
6561 \setcounter{LWR@autoindex}{0}
6562
6563 \newcounter{LWR@autoglossary}
6564 \setcounter{LWR@autoglossary}{0}

```

`\printindex`

```

6565 \let\LWR@origprintindex\printindex
6566
6567 \renewcommand*\printindex
6568 {
6569 \LWR@startpars
6570 \LWR@origprintindex
6571 }

```

Env `theindex`

```

6572 \@ifundefined{chapter}
6573 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}
6574 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}
6575
6576 \renewenvironment*{theindex}{%
6577 \LWR@indexsection{indexname}%
6578 \let\item\LWR@indexitem%
6579 \let\subitem\LWR@indexsubitem%
6580 \let\subsubitem\LWR@indexsubsubitem%
6581 }{}

```

`\LWR@indexitem`

```

6582 \newcommand*\LWR@indexitem){
6583
6584 \InlineClass{indexitem}{}
6585 }

```

`\LWR@indexitem`

```

6586 \newcommand*\LWR@indexsubitem){
6587
6588 \InlineClass{indexsubitem}{}
6589 }

```

`\LWR@indexitem`

```

6590 \newcommand*\LWR@indexsubsubitem){
6591
6592 \InlineClass{indexsubsubitem}{}
6593 }

```

`\@wrindex` `{\langle term \rangle}` Redefined to write the `LWR@latestautopage` counter instead of page

```

6594 \def\LWR@wrindex#1{%
6595 \addtocounter{LWR@autoindex}{1}%
6596 \LWR@newlabel{LWRindex-\theLWR@autoindex}%
6597 \protected@write\@indexfile{}%
6598 {\string\indexentry{#1}{\theLWR@autoindex}}%
6599 \endgroup
6600 \@esphack}
6601
6602 \let\@wrindex\LWR@wrindex

```

`\@wrglossary` `{\langle term \rangle}` Redefined to write the `LWR@latestautopage` counter instead of page

```

6603 \def\@wrglossary#1{%
6604 \addtocounter{LWR@autoglossary}{1}%
6605 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
6606 \protected@write\@glossaryfile{%
6607 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
6608 \endgroup
6609 \@esphack}

```

`\hyperindexref` `{\langle autosecnumber \rangle}`

`\hyperindexref{web address}` is inserted into *.ind by the xindy style file lwarp.xdy

```

6610 \newcommand*\hyperindexref[1]{\nameref{LWRindex-#1}}
6611 \end{warpHTML}

```

for PRINT output: A null command for print mode, in case hyperref was not used:

```

6612 \begin{warpprint}
6613 \newcommand{\hyperindexref}[1]{#1}
6614 \end{warpprint}

```

for HTML & PRINT: For the glossaries package, try to prevent an error where `\glo@name` was not found:

```

6615 \begin{warpall}
6616 \providecommand{\glo@name}{}
6617 \end{warpall}

```

66 Restoring original formatting

`\LWR@restoreorigformatting` Used to temporarily restore the print-mode meaning of a number of formatting, graphics, and symbols-related macros while generating svg math or a lateximage. A number of packages will `\appto` additional actions to this macro.

Various packages add to this macro using `\appto`.

```

6618 \newcommand*\LWR@restoreorigformatting{%
6619 \LWR@traceinfo{LWR@restoreorigformatting}%
6620 \linespread{1}%

6621 \LetLtxMacro\ref\LWR@origref%{} syntax highlighting
6622 \RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
6623 \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
6624 \renewcommand{\BlockClassSingle}[2]{##2}%

```

```
6625 \LetLtxMacro{\hspace}{\LWR@origspace}%
6626 \LetLtxMacro\rule\LWR@origrule%
6627 \let\,\LWR@origcomma% disable HTML short unbreakable space
6628 \let\textellipsis\LWR@origtextellipsis%
6629 \let\textless\LWR@origtextless%
6630 \let\textgreater\LWR@origtextgreater%
6631 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
6632 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
6633 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
6634 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
6635 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
6636 \LetLtxMacro{\textit}{\LWR@origtextit}%
6637 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
6638 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
6639 \LetLtxMacro{\textup}{\LWR@origtextup}%
6640 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
6641 \LetLtxMacro{\emph}{\LWR@origemph}%
6642 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
6643 \LetLtxMacro{\sffamily}{\LWR@origsfamily}%
6644 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
6645 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
6646 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
6647 \LetLtxMacro{\upshape}{\LWR@origupshape}%
6648 \LetLtxMacro{\slshape}{\LWR@origslshape}%
6649 \LetLtxMacro{\scshape}{\LWR@origscshape}%
6650 \LetLtxMacro{\itshape}{\LWR@origitshape}%
6651 \LetLtxMacro{\em}{\LWR@origem}%
6652 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
6653 \let\sp\LWR@origsp%
6654 \let\sb\LWR@origsb%
6655 \LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
6656 \LetLtxMacro\@textsuperscript\LWR@orig@textsuperscript%
6657 \LetLtxMacro\textsubscript\LWR@origtextsubscript%
6658 \LetLtxMacro\@textsubscript\LWR@orig@textsubscript%
6659 \LetLtxMacro\underline\LWR@origunderline%
6660 \let~\LWR@origtilde%
6661 \let\enskip\LWR@origenskip%
6662 \let\quad\LWR@origquad%
6663 \let\qquad\LWR@origqquad%
6664 \LetLtxMacro{\tabular}{\LWR@origtabular}%
6665 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
6666 \LetLtxMacro\toprule\LWR@origtoprule%
6667 \LetLtxMacro\midrule\LWR@origmidrule%
6668 \LetLtxMacro\cmidrule\LWR@origcmidrule%
6669 \LetLtxMacro\bottomrule\LWR@origbottomrule%
6670 \LetLtxMacro\addlinespace\LWR@origaddlinespace%
6671 \LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
6672 \LetLtxMacro\specialrule\LWR@origspecialrule%
6673 \let\newline\LWR@orignewline%
6674 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
```

```

6675 \LetLtxMacro\includegraphics\LWR@originincludegraphics%
6676 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
6677 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
6678 \let\reflectbox\LWR@origreflectbox%
6679 \LetLtxMacro\resizebox\LWR@origresizebox%
6680 \let\framebox\LWR@origframebox%
6681 \let\makebox\LWR@origmakebox%
6682 \let\fbbox\LWRprint@fbbox%
6683 \let\fbboxBlock\LWRprint@fbbox%
6684 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
6685 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
6686 \LetLtxMacro{\minipage}{\LWR@origminipage}%
6687 \let\endminipage\LWR@endminipage%
6688 \LetLtxMacro{\parbox}{\LWR@origparbox}%
6689 \let\TeX\LWR@origTeX%
6690 \let\LaTeX\LWR@origLaTeX%
6691 \let\LaTeXe\LWR@origLaTeXe%
6692 \renewcommand*{\Xe}{X\textsubscript{E}}%

6693 \LetLtxMacro\@ensuredmath\LWR@origensuredmath%

6694 \csletcs{equation*}{LWR@origequationstar}%
6695 \csletcs{endequation*}{LWR@origendequationstar}%
6696 %
6697 \LWR@restoreorigaccents%
6698 \LWR@restoreoriglists%
6699 %
6700 \LWR@FBcancel%
6701 }

```

67 Math

67.1 Limitations

67.1.1 Rendering tradeoffs

Math rendering Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.

SVG files In its current implementation, rendering math as images creates a new SVG file for each expression. 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 check-summing may be used to remove the need for duplicate files.

- SVG inline** Another approach could be to in-line the SVG files directly into the HTML. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.
- PNG files** Others converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are the preferred approach for scalable graphics.
- MathML** Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 9 regarding EPUB output with MathJax.

67.1.2 SVG option

- SVG math option** For SVG math, math is rendered as usual by \TeX 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 \TeX 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 \TeX code which generated the math, allowing copy/paste of the \TeX 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 math copy/paste** For SVG math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the \TeX code for the math expression. For \mathcal{AMS} environments with multiple numbers in the same environment, only the first and last is copy/pasted, as a range. No tags are listed inside a starred \mathcal{AMS} environment, although the `\tag` macro will still appear inside the \TeX math expression.

67.1.3 MathJax option

MathJax math option The popular MathJax alternative (mathjax.org) may be used to display math.

Prog MathJax

When MathJax is enabled, math is rendered twice:

1. As regular \TeX PDF output placed inside an HTML comment, allowing equation numbering and cross referencing to be almost entirely under the control of \TeX , and

¹³See section 273 regarding fonts and fractions.

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

MathJax limitations

Prog MathJax

Limitations when using MathJax include:

chapter numbers

- In document classes which have chapters, `\tagged` equations have the chapter number prepended in HTML output, unlike \LaTeX . `\tag*` equations (correctly) do not. This may be improved with future versions of the MathJax support script.

<https://groups.google.com/forum/#!topic/mathjax-users/jUtewUcE2bY>

subequations

- MathJax itself does not support subequations. This may be improved by parsing the \LaTeX math expression to manually insert tags, but this has not yet been done.

footnotes in math

- Footnotes inside equations are not yet supported while using MathJax.

lateximage

- Math appearing inside a `lateximage`, and therefore also inside a `Tikz` or `picture` environment, is rendered as SVG math even if MathJax is used in the rest of the document.

siunitx

- Usage of `siunitx` inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.

<https://github.com/burnpanck/MathJax-siunitx>

Also see section 8.5.5.

⚠ `siunitx` inside an equation

⚠ other macros and packages

- Other math-related macros and packages are not supported by MathJax, including `\ensuremath`, `bigdelim`, `units`, and `nicefrac`, along with occasionally-used macros such as `\footnote` and `\relax`.

custom MathJax macros

- MathJax does not automatically support custom \LaTeX macros, but they may be created by the user inside a math expression:

```

\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}

```

67.2 Inline and display math

for HTML output: 6702 `\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.

```

6703 \let\LWR@origtextdollar\$
6704
6705 \renewcommand*{\$}{\%
6706 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
6707 {\LWR@origtextdollar}%
6708 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
6709 }

```

`Ctr LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```

6710 \newcounter{LWR@externalfilecnt}

6711 \LetLtxMacro\LWR@origdollar$
6712 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

6713 \LetLtxMacro\LWR@origopenparen\<
6714 \LetLtxMacro\LWR@origcloseparen\
6715 \LetLtxMacro\LWR@origopenbracket\[
6716 \LetLtxMacro\LWR@origclosebracket\

```

$\$$ Redefine the dollar sign to place math inside a lateximage, or use MathJax:
 $\$\$$

```
6717 \begingroup
6718 \catcode'\$=\active%
6719 \protected\gdef$\{\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

$\LWR@doubledollar$ Redefine the double dollar sign to place math inside a lateximage, or use MathJax:

```
6720 \protected\gdef\LWR@doubledollar$#1$\${
```

If MathJax or formatting for a word processor, print the \LaTeX expression:

```
6721 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between \lbrack and \rbrack :

```
6722 {
6723
6724     \textbackslash[%
6725     \LWR@HTMLsanitize{#1}%
6726     \textbackslash]
6727
6728 }% mathjax
```

For SVG, print the math inside a lateximage, with an `<alt>` tag of the \LaTeX code:

```
6729 {% not mathjax
6730
6731     \begin{lateximage}%
6732     [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
6733     \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
6734     \end{lateximage}%
6735
6736 }% not mathjax
6737 }%
```

$\LWR@singledollar$ Redefine the single dollar sign to place math inside a lateximage, or use MathJax:

```
6738 \protected\gdef\LWR@subsingledollar#1{%
6739 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MathJax, print the math between \lrcorner and \rcom :

```
6740 {%
6741     {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}%
6742 }% mathjax
```

For SVG, print the math inside a lateximage, with an <alt> tag of the \LaTeX code:

```

6743 {% not mathjax
6744     \begin{lateximage}%
6745     [\textbackslash( \LWR@HTMLSanitize{#1} \textbackslash)]%
6746     \LWR@origdollar#1\LWR@origdollar%
6747     \end{lateximage}%
6748 }%
6749 }
6750
6751 \protected\gdef\LWR@singledollar#1${%
6752 \LWR@subsingledollar{#1}%
6753 }

```

\(Redefine to the above dollar macros.

```

\
6754 \protected\gdef\(#1)\{%
6755 $#1$}
6756 \protected\gdef\[#1\]{$$#1$$}
6757
6758 \endgroup
6759
6760
6761

```

\@ensuredmath {<expression>} Not yet adapted to lwarp.

```

6762 \letLtxMacro\LWR@origensuredmath\@ensuredmath
6763
6764 \renewcommand{\@ensuredmath}[1]{%
6765 \ifmmode%
6766     \LWR@origensuredmath{#1}%
6767 \else%
6768     \LWR@subsingledollar{\relax#1}%
6769 \fi%
6770 }

```

Remove the old math and displaymath environments:

```

6771 \let\math\relax
6772 \let\endmath\relax
6773 \let\displaymath\relax
6774 \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`.

```
6775 \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`.

```
6776 \NewEnviron{displaymath}{\expandafter[\BODY\]\@ignoretrue}
```

67.3 MathJax support

Ctrl `LWR@nextequation` Used to add one to compute the next equation number.

```
6777 \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 \TeX .$$

```
6778 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
6779 \ifcsdef{thechapter}{
6780 \InlineClass{hidden}{
6781 \textbackslash(
6782 \textbackslash{)seteqsection \{thechapter\}
6783 \textbackslash)
6784 }
6785 }
6786 {}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
6787 \ifthenelse{\cnttest{\value{equation}}>0}
6788 {
```

Tell MathJax that the next set of equations begins with the current \TeX equation number, plus one.

```
6789 \setcounter{LWR@nextequation}{\value{equation}}
6790 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\ (“ and “\)” characters, to be printed to HTML, not interpreted by \LaTeX .

```
6791 \InlineClass{hidden}{
6792   \textbackslash(
6793     \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
6794   \textbackslash)
6795 }
6796 }{}% not eq > 0
6797 }
```

`\LWR@hidelatexequation` $\langle environment \rangle$ $\langle contents \rangle$

Creates the \LaTeX version of the equation inside an HTML comment.

```
6798 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
6799 \LWR@stoppars
6800 \LWR@htmlopencomment
6801
```

Start the \LaTeX math environment inside the HTML comment:

```
6802 \begingroup
6803 \csuse{LWR@orig#1}
```

While in the math environment, restore various commands to their \LaTeX meanings.

```
6804 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 67.5.1.

Print the contents of the equation:

```
6805 #2
```

End the \LaTeX math environment inside the HTML comment:

```
6806 \csuse{LWR@origend#1}
6807 \endgroup
6808
```

Close the HTML comment and resume HTML paragraph handling:

```
6809 \LWR@htmlclosecomment
6810 \LWR@startpars
```

6811 }

`\LWR@addmathjax` $\langle environment \rangle$ $\langle contents \rangle$

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 .

6812 `\NewDocumentCommand{\LWR@addmathjax}{m +m}{%`

Enclose the MathJax environment inside printed “\ (“ and “\)” characters.

6813 `\LWR@origtilde\LWR@orignewline`

6814 `\textbackslash{}begin\{#1\}`

Print the contents, sanitizing for HTML special characters.

6815 `\LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}`

Close the MathJax environment:

6816 `\textbackslash{}end\{#1\}`

6817 `\LWR@orignewline`

6818 }

67.4 Equation environment

Remember existing equation environment:

6819 `\let\LWR@origequation\equation`

6820 `\let\LWR@origendequation\endequation`

Remove existing equation environment:

6821 `\let\equation\relax`

6822 `\let\endequation\relax`

`Env` `equation` The new equation environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original equation inside a `lateximage`, along with an `<alt>` tag containing a detokenized copy of the \LaTeX source for the math.

For MathJax output, the contents are typeset in an original equation environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
6823 \NewEnviron{equation}{%
6824
```

If `mathjax` or `FormatWP`, print the \TeX expression:

```
6825 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

MathJax output:

```
6826 {
```

Print commands to synchronize MathJax's equation number and format to the current \TeX chapter/section and equation number:

```
6827 \LWR@syncmathjax
```

Print the \TeX math inside an HTML comment:

```
6828 \LWR@hidelatexequation{equation}{\BODY}
6829 }
```

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`.

```
6830 {% not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
6831 \begin{lateximage}[({\LWR@equationtag} \textbackslash{begin}\{equation\}) %
6832 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}} %
6833 \textbackslash{end}\{equation\}}]% alt tag
```

Create the actual \TeX -formatted equation inside the `lateximage` using the contents of the environment.

```
6834 \LWR@origequation
6835 \BODY% contents collected by NewEnviron
6836 \LWR@origendequation
6837 \end{lateximage}%
6838 }% not mathjax
6839
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
6840 }[%
6841   \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6842   {%
6843     \LWR@addmathjax{equation}{\BODY}%
6844   }{%
6845
6846 ]
```

Env `equation*`

```
6847 \csletcs[LWR@origequationstar]{equation*}
6848 \csletcs[LWR@origendequationstar]{endequation*}
6849 \renewenvironment*{equation*}
6850 {\displaymath}
6851 {\enddisplaymath}
```

67.5 AMS Math environments

67.5.1 Support macros

Bool `LWR@amsmultiline` True if processing a multiline environment.

To compensate for `multiline`-specific code, `LWR@amsmultiline` is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an `amsmath` environment which is not a `multiline` environment and not an equation.

```
6852 \newbool{LWR@amsmultiline}
6853 \boolfalse{LWR@amsmultiline}
```

`\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`.

```
6854 \newcommand*{\LWR@htmlmathlabel}[1]{%
6855 \LWR@traceinfo{LWR@htmlmathlabel #1}%
```

If `mathjax` or `FormatWP`, print the \LaTeX expression:

```
6856 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6857 {%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
6858   \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6859   \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.)

```
6860   \LWR@htmlclosecomment%
6861   \LWR@origltx@label{#1}%
6862   \LWR@htmlopencomment%
6863   }% text
6864 }% mathjax
6865 {%
6866   \LWR@origltx@label{#1}%
6867 }%
6868 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
6869 \newcommand*{\LWR@beginhideamsmath}{
6870 \LWR@stoppars
6871 \LWR@origtilde\LWR@orignewline
6872 \LWR@htmlopencomment
6873
6874 \begingroup
6875 \LWR@restoreorigformatting
6876 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```
6877 \newcommand*{\LWR@endhideamsmath}{
6878 \endgroup
6879
6880 \LWR@htmlclosecomment
6881 \LWR@orignewline
6882 \LWR@startpars
6883 }
```

67.5.2 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 optional argument of `\LWR@amsmathbody` or `\LWR@amsmathbodynumbered` telling `lateximage` to use as the HTML `<alt>` tag the environment's contents which were automatically captured by the \mathcal{AMS} environment.

For MathJax: Each environment is syched with \TeX 's equation numbers, typeset with \TeX inside an HTML comment, then printed to HTML output for MathJax to process.

Env `eqnarray` This environment 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`.

```
6884 \let\LWR@origeqnarray\eqnarray
6885 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6886 \newbool{LWR@numbereqnarray}
6887 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
6888 \newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the \TeX expression:

```
6889 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6890 {%
```

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:

```
6891 \LWR@syncmathjax
6892 \boolfalse{LWR@amsmultiline}
6893 \ifbool{LWR@numbereqnarray}
6894 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```

6895     \LWR@beginhideamsmath
6896     \LWR@origeqnarray
6897     \BODY
6898     \LWR@origendeqnarray
6899     \LWR@endhideamsmath

```

Then print the (sanitized) contents to the output for MathJax to interpret:

```

6900     \LWR@addmathjax{eqnarray}{\BODY}
6901     }%
6902     {% not LWR@numbreqnarray

```

If not numbering equations, just create the contents for MathJax:

```

6903     \LWR@addmathjax{eqnarray*}{\BODY}
6904     }% LWR@numbreqnarray
6905     }% mathjax
6906     {% not mathjax
6907     \ifbool{LWR@numbreqnarray}
6908     {

```

For numbered svg equations, first create a lateximage with an alt attribute containing sanitized copy of the source code:

```

6909     \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
6910     \LWR@addmathjax{eqnarray}{\BODY}]

```

Then create the image contents using an actual eqnarray:

```

6911     \LWR@origeqnarray
6912     \BODY
6913     \LWR@origendeqnarray
6914     \end{lateximage}
6915     }%
6916     {% not LWR@numbreqnarray

```

If not numbered, do the same, but an extra \nonumber seems to be required:

```

6917     \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
6918     \LWR@origeqnarray
6919     \BODY
6920     \nonumber
6921     \LWR@origendeqnarray
6922     \end{lateximage}
6923     }% LWR@numbreqnarray
6924     }% not mathjax

```

Default to number equations in the future:

```
6925 \booltrue{LWR@numbereqnarray}
6926 }
```

eqnarray itself is made with a blank line before and after to force it to be on its own line:

```
6927 \RenewEnviron{eqnarray}
6928 {%
6929
6930 \LWR@eqnarrayfactor
6931
6932 }
```

The starred version is patched to turn off the numbering:

```
6933 \csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

The following $\mathcal{A}\mathcal{M}\mathcal{S}$ environments are more easily patched in-place:

Env `multline`

```
6934 \BeforeBeginEnvironment{multline}{
6935
6936 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6937 {
6938     \LWR@syncmathjax
6939     \booltrue{LWR@amsmultline}
6940     \LWR@beginhideamsmath
6941 }
6942 {
6943     \begin{lateximage}[\LWR@amsmathbodynumbered{multline}]
6944 }
6945 }
6946
6947 \AfterEndEnvironment{multline}{
6948
6949 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6950 {
6951     \LWR@endhideamsmath
6952     \boolfalse{LWR@amsmultline}
6953     \LWR@addmathjax{multline}{\the\@envbody}
6954 }
6955 {\end{lateximage}}
6956
6957 }
```

Env `multline*`

```
6958 \BeforeBeginEnvironment{multline*}{
6959
6960 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6961 {
6962   \LWR@syncmathjax
6963   \booltrue{LWR@amsmultline}
6964   \LWR@beginhideamsmath
6965 }
6966 {
6967   \begin{lateximage}[\LWR@amsmathbody{multline*}]
6968 }
6969 }
6970
6971 \AfterEndEnvironment{multline*}{
6972
6973 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6974 {
6975   \LWR@endhideamsmath
6976   \boolfalse{LWR@amsmultline}
6977   \LWR@addmathjax{multline*}{\the\@envbody}
6978 }
6979 {\end{lateximage}}
6980
6981 }
6982
```

Env **gather**

```
6983 \BeforeBeginEnvironment{gather}{
6984
6985 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6986 {
6987   \LWR@syncmathjax
6988   \boolfalse{LWR@amsmultline}
6989   \LWR@beginhideamsmath
6990 }
6991 {
6992   \begin{lateximage}[\LWR@amsmathbodynumbered{gather}]
6993 }
6994 }
6995
6996 \AfterEndEnvironment{gather}{
6997
6998 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
6999 {
7000   \LWR@endhideamsmath
7001   \LWR@addmathjax{gather}{\the\@envbody}
7002 }
7003 {\end{lateximage}}
```

```
7004
7005 }
```

Env **gather***

```
7006 \BeforeBeginEnvironment{gather*}{
7007
7008 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7009 {
7010   \LWR@syncmathjax
7011   \boolfalse{LWR@amsmultline}
7012   \LWR@beginhideamsmath
7013 }
7014 {
7015   \begin{lateximage}[\LWR@amsmathbody{gather*}]
7016 }
7017 }
7018
7019 \AfterEndEnvironment{gather*}{
7020
7021 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7022 {
7023   \LWR@endhideamsmath
7024   \LWR@addmathjax{gather*}{\the\@envbody}
7025 }
7026 {\end{lateximage}}
7027
7028 }
```

Env **align**

```
7029 \BeforeBeginEnvironment{align}{
7030
7031 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7032 {
7033   \LWR@syncmathjax
7034   \boolfalse{LWR@amsmultline}
7035   \LWR@beginhideamsmath
7036 }
7037 {
7038   \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
7039 }
7040 }
7041
7042 \AfterEndEnvironment{align}{
7043
7044 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7045 {
```

```
7046 \LWR@endhideamsmath
7047 \LWR@addmathjax{align}{\the\@envbody}
7048 }
7049 {\end{lateximage}}
7050
7051 }
```

Env `align*`

```
7052 \BeforeBeginEnvironment{align*}{
7053
7054 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7055 {
7056 \LWR@syncmathjax
7057 \boolfalse{LWR@amsmultline}
7058 \LWR@beginhideamsmath
7059 }
7060 {
7061 \begin{lateximage}[\LWR@amsmathbody{align*}]
7062 }
7063 }
7064
7065 \AfterEndEnvironment{align*}{
7066
7067 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7068 {
7069 \LWR@endhideamsmath
7070 \LWR@addmathjax{align*}{\the\@envbody}
7071 }
7072 {\end{lateximage}}
7073
7074 }
```

Env `flalign`

```
7075 \BeforeBeginEnvironment{flalign}{
7076
7077 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7078 {
7079 \LWR@syncmathjax
7080 \boolfalse{LWR@amsmultline}
7081 \LWR@beginhideamsmath
7082 }
7083 {
7084 \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
7085 }
7086 }
7087 }
```

```

7088 \AfterEndEnvironment{flalign}{
7089
7090 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7091 {
7092   \LWR@endhideamsmath
7093   \LWR@addmathjax{flalign}{\the\@envbody}
7094 }
7095 {\end{lateximage}}
7096
7097 }

```

Env `flalign*`

```

7098 \BeforeBeginEnvironment{flalign*}{
7099
7100 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7101 {
7102   \LWR@syncmathjax
7103   \boolfalse{LWR@amsmultiline}
7104   \LWR@beginhideamsmath
7105 }
7106 {
7107   \begin{lateximage}[\LWR@amsmathbody{flalign*}]
7108 }
7109 }
7110
7111 \AfterEndEnvironment{flalign*}{
7112
7113 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7114 {
7115   \LWR@endhideamsmath
7116   \LWR@addmathjax{flalign*}{\the\@envbody}
7117 }
7118 {\end{lateximage}}
7119
7120 }

7121 \end{warpHTML}

```

68 Lateximages

68.1 Description

Env `lateximage` A `lateximage` is a piece of the document which is typeset in \TeX then included in

the HTML output as an image. This is used for math if `svg math` is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 66.)

A `lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceding page and ends on following page, and instructions are written to `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 `` 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}
```

68.2 Support counters and macros

for HTML output: 7122 `\begin{warpHTML}`

Ctrl `LWR@lateximagenumber` Sequence the images.

```
7123 \newcounter{LWR@lateximagenumber}
7124 \setcounter{LWR@lateximagenumber}{0}
```

Ctrl `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
7125 \newcounter{LWR@lateximagedepth}
7126 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
7127 \edef\LWR@hashmark{\string#} % for use in \write
7128 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctrl `LWR@LIpage` Used to reference the PDF page number of a `lateximage` to be written into `lateximages.txt`.

```
7129 \newcounter{LWR@LIpage}
```

```
7130 \end{warpHTML}
```

68.3 Font size

for HTML & PRINT: 7131 \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.

 **no backslash** *Do not include the leading backslash in the name.*

```
7132 \newcommand*{\LateximageFontSizeName}{large}
```

```
7133 \end{warpall}
```

68.4 Sanitizing math expressions for HTML

for HTML output: 7134 \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...

```
7135 \newcommand{\LWR@HTMLsanitize}[1]{%
7136 \begingroup%
7137 \LWR@FBcancel%
7138 \protect\StrSubstitute{\detokenize{#1}}%
7139 {\detokenize{&}}%
7140 {\detokenize{&}}[\LWR@strresult]%
7141 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7142 {\detokenize{<}}%
7143 {\detokenize{<}}[\LWR@strresult]%
7144 [\LWR@strresult]%
7145 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7146 {\detokenize{>}}%
7147 {\detokenize{>}}[\LWR@strresult]%
7148 [\LWR@strresult]%
7149 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7150 {\detokenize{##}}%
```

```

7151 {\#}%
7152 [\LWR@strresult]%
7153 \LWR@strresult%
7154 \endgroup%
7155 }

```

`\LWR@HTMLsanitizeexpand` $\{(\textit{text})\}$

This version expands the argument before sanitizing it.

```

7156 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
7157 \begingroup%
7158 \LWR@FBcancel%
7159 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
7160 {\detokenize{&}}%
7161 {\detokenize{&};}%
7162 [\LWR@strresult]%
7163 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7164 {\detokenize{<}}%
7165 {\detokenize{&lt;};}%
7166 [\LWR@strresult]%
7167 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7168 {\detokenize{>}}%
7169 {\detokenize{&gt;};}%
7170 [\LWR@strresult]%
7171 \LWR@strresult%
7172 \endgroup%
7173 }

```

68.5 Equation numbers

`\LWR@startingequation` For use with `lateximage` and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```

7174 \newcounter{LWR@startingequation}
7175
7176 \@ifundefined{chapter}
7177 {
7178 \renewcommand{\theLWR@startingequation}{%
7179 \arabic{LWR@startingequation}%
7180 }
7181 }
7182 {% chapter defined
7183 \renewcommand{\theLWR@startingequation}{%
7184 \ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}}%
7185 \arabic{LWR@startingequation}%

```

```
7186 }
7187 }
```

Bool True for the first equation tag, false for later tags in the same environment.
`LWR@isstartingequation`

```
7188 \newbool{LWR@isstartingequation}
```

`\LWR@startingequationtag` Prints the starting equation number or tag.

```
7189 \let\LWR@startingequationtag\theLWR@startingequation
```

`\LWR@equationtag` Prints the ending equation number or tag.

```
7190 \let\LWR@equationtag\theequation
```

Only if svg math, patch `\tag` after packages have loaded, in case someone else modified `\tag`.

```
7191 \AtBeginDocument{
7192
7193 \ifbool{mathjax}{}{% not mathjax
```

`\LWR@remembertag` $\langle tag \rangle$

For use inside the math environments while using svg math. Sets `\theLWR@startingequation` and `\theequation` to the given tag.

```
7194 \NewDocumentCommand{\LWR@remembertag}{m}{%
7195 \ifbool{LWR@isstartingequation}%
7196 {%
7197   \global\boolfalse{LWR@isstartingequation}%
7198   \xdef\LWR@startingequationtag{#1}%
7199 }%
7200 }%
7201 \xdef\LWR@equationtag{#1}%
7202 }%
```

Patches for \mathcal{AMS} math `\tag` macro to remember the first tag:

```
7203 \LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
7204 \LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@
7205
7206 \renewcommand*{\make@df@tag@@}[1]{%
7207 \LWR@remembertag{#1}%
7208 \LWR@origmake@df@tag@@{#1}%
```

```

7209 }
7210
7211 \renewcommand*\make@df@tag@@@[1]{%
7212 \LWR@remember@tag{#1}%
7213 \LWR@origmake@df@tag@@@{#1}%
7214 }
7215
7216 }% not mathjax
7217 }% AtBeginDocument

```

68.6 HTML <alt> tags

`\LWR@amsmathbody` $\langle envname \rangle$ For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag.

```

7218 \newcommand*\LWR@amsmathbody}[1]
7219 {%
7220 \textbackslash\begin\}\{#1\} %
7221 \LWR@HTMLsanitizeexpand{\detokenize\expandafter\the\@envbody}}%
7222 \textbackslash\end\}\{#1\}%
7223 }

```

`\LWR@amsmathbodynumbered` $\langle envname \rangle$ For use inside the optional argument to a `lateximage` to add the contents of a AMS math environment to the `<alt>` tag, prefixed by the equation numbers.

```

7224 \newcommand*\LWR@amsmathbodynumbered}[1]
7225 {%
7226 \ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}%
7227 {(\LWR@equationtag)}%
7228 {(\LWR@startingequationtag--\LWR@equationtag)} %
7229 \LWR@amsmathbody{#1} %
7230 }

```

68.7 lateximage

Env `lateximage` [$\langle alt \rangle$ tag]

```

7231 \catcode'\$=\active%
7232
7233
7234 \NewDocumentEnvironment{lateximage}{0{image}}
7235 {%
7236 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%

```

Nested lateximages remain one large lateximage:

```
7237 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
7238 {%
7239   \addtocounter{LWR@lateximagedepth}{1}%
7240 }%
```

Otherwise, this is the outer-most lateximage:

```
7241 {% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
7242   \setcounter{LWR@startingequation}{\value{equation}}%
7243   \addtocounter{LWR@startingequation}{1}%
7244   \booltrue{LWR@isstartingequation}%
7245   \let\LWR@startingequationtag\theLWR@startingequation%
7246   \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
7247   \addtocounter{LWR@lateximagenumber}{1}%
7248   \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
7249   \boolfalse{mathjax}%
```

Be sure that are doing a paragraph:

```
7250   \LWR@ensuredoingapar%
```

Next file:

```
7251   \addtocounter{LWR@externalfilecnt}{1}%
7252   \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
7253   \setcounterpageref{LWR@LIpage}{LWR@lateximage\arabic{LWR@lateximagenumber}}%
7254   \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:

```
7255 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
7256 class="lateximagesource"{}} \LWR@orignewline
```

Write instructions to the lateximages.txt file:

```
7257 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
7258 \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.

```
7259 \LWR@traceinfo{lateximage: about to create open comment}%
7260 \LWR@htmlopencomment%
```

One level deeper:

```
7261 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
7262 \LWR@traceinfo{lateximage: about to create a new page}%
7263 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
7264 \LWR@traceinfo{lateximage: about to create minipage}%
7265 \LWR@origminipage{6in}%
7266 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```
7267 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
7268 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
7269 \def\@mpfn{footnote}%
7270 \def\thempfn{\thefootnote}%
7271 \let\@footnotetext\LWR@footnotetext%
```

Create the LWRlateximage<number> label:

```

7272 \LWR@traceinfo{lateximage: about to create label}%
7273 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
7274 \LWR@traceinfo{lateximage: finished creating the label}%

```

Enable print-mode math functions:

```

7275 \LetLtxMacro$\LWR@origdollar%
7276 \catcode'\$=3% math shift
7277 \LetLtxMacro\(\LWR@origopenparen%
7278 \LetLtxMacro\)\LWR@origcloseparen%
7279 \LetLtxMacro\[\LWR@origopenbracket%
7280 \LetLtxMacro\]\LWR@origclosebracket%
7281 }% end of outer-most lateximage
7282 \LWR@traceinfo{lateximage: finished start of environment}%
7283 }% end of \begin{lateximage}

```

`\endlateximage` When the environment closes:

```

7284 {% start of \end{lateximage}
7285 \LWR@traceinfo{lateximage: starting end of environment}%

```

Nested more than one deep?

```

7286 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

7287 {\addtocounter{LWR@lateximagedepth}{-1}}%

```

If this is the outer-most lateximage:

```

7288 {% end of outer-most lateximage

```

Finish the lateximage minipage and start a new PDF page:

```

7289 \LWR@origendminipage%
7290 \LWR@orignewpage%
7291 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```

7292 \LWR@htmlclosecomment{\LWR@orignewline%
7293 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%

```

Create a link to the lateximage, allowing its natural height:

```
7294 \LWR@subinlineimage[#1]{lateximage}%
7295 {\lateximages\OSPathSymbol{}}\lateximage-\theLWR@externalfilecnt}{svg}{}
```

Be sure that are doing a paragraph:

```
7296 \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```
7297 \LWR@htmltag{/span}%
7298 \ifbool{HTMLDebugComments}{%
7299 \LWR@htmlcomment{End of lateximage}%
7300 }{}%
7301 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
7302 \addtocounter{LWR@lateximagedepth}{-1}%
7303 }% end of outer-most lateximage
7304 \LWR@traceinfo{lateximage: done}%
7305 }%
7306 \catcode'\$=3% math shift
7307 \end{warpHTML}
```

```
for PRINT output: 7308 \begin{warpprint}
7309 % \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
7310 \newenvironment{lateximage}[1][{}]{}
7311 \end{warpprint}
```

69 center, flushleft, flushright

```
for HTML output: 7312 \begin{warpHTML}
```

Env center Replace center functionality with css tags:

```
7313 \renewenvironment*{center}
7314 {
7315 \LWR@forcenewpage
7316 \ifbool{FormatWP}
7317 {\BlockClass[text-align:center]{center}}
7318 {\BlockClass{center}}
7319 }
7320 {\endBlockClass}
```

Env `flushright`

```
7321 \renewenvironment*{flushright}
7322 {
7323 \LWR@forcenewpage
7324 \ifbool{FormatWP}
7325 {\BlockClass[text-align:right]{flushright}}
7326 {\BlockClass{flushright}}
7327 }
7328 {\endBlockClass}
```

Env `flushleft`

```
7329 \renewenvironment*{flushleft}
7330 {
7331 \LWR@forcenewpage
7332 \ifbool{FormatWP}
7333 {\BlockClass[text-align:left]{flushleft}}
7334 {\BlockClass{flushleft}}
7335 }
7336 {\endBlockClass}

7337 \end{warpHTML}
```

70 Pre-loaded packages

for HTML output: 7338 `\begin{warpHTML}`

If `textcomp` was loaded before `lwarp`, perhaps as part of the font-related packages, explicitly load the `lwarp` patches now:

```
7339 \@ifpackageloaded{textcomp}
7340 {
7341 \LWR@origRequirePackage{lwarp-textcomp}
7342 }
7343 {}
```

If `graphics` or `graphicx` were loaded before `lwarp`, perhaps by `xunicode`, explicitly load the `lwarp` patches now:

```
7344 \@ifpackageloaded{graphics}
7345 {
7346 \LWR@origRequirePackage{lwarp-graphics}
7347 }
7348 {}
```

```
7349 \end{warpHTML}
```

71 Siunitx

`Pkg siunitx` The lwarp core passes a few options to siunitx.

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

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

NOTE: As of this writing, the `siunitx` extension for MathJax is not currently hosted at any public CDN, thus `siunitx` is not usable with MathJax unless a local copy of this extension is created first.

for HTML output: `7350 \begin{warpHTML}`

Options for siunitx:

```
7351 \PassOptionsToPackage{
7352   detect-mode=true,
7353   per-mode=symbol,% fraction is not seen by pdftotext
7354 %   text-celsius = {\protect\LWRsiunitx@degree}{C},
7355 %   text-degree = {\protect\LWRsiunitx@degree},
7356 }{siunitx}
7357
```

```
7358 \end{warpHTML}
```

72 Graphics print-mode modifications

72.1 General limitations

- ⚠ **.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 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 CSS3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike L^AT_EX, so expect some ugly results for scaling and rotating.

72.2 Print-mode modifications

for PRINT output: For print output, accept and then discard the new class key:

```
7359 \begin{warpprint}
7360 \define@key{Gin}{class}{}
7361 \end{warpprint}
```

Print-mode additions for the overpic package. See section 202 for the HTML version.

```
7361 \AtBeginDocument{
7362 \@ifpackageloaded{overpic}{
7363 \newcommand*\overpicfontsize{12}
7364 \newcommand*\overpicfontskip{14}
7365 }{}
```

```
7366 }
7367 \end{warpprint}
```

73 Xcolor boxes

Pkg xcolor A few new definitions are provided for enhanced HTML colored boxes, and `\fcolorbox` is slightly modified. Print-mode version are also provided.

Print-mode versions of new xcolor defintions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the xcolor originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of none, in which case only the frame is drawn, allowing the background page color to show.

for HTML & PRINT: 7368 `\begin{warpall}`

After xparse may have been loaded ...

```
7369 \AtBeginDocument{
```

... and *only* if xcolor was loaded:

```
7370 \@ifpackageloaded{xcolor}{
7371 \LWR@traceinfo{patching xcolor}
```

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

```
7372 \LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when xcolor is loaded. Following is the print-mode action:

```
7373 \warpprintonly{
7374 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
7375 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
7376 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
7377 }
```

`\fcolorbox` [`\framemodel`] [`\framecolor`] [`\boxmodel`] [`\boxcolor`] [`\text`]

In print mode, `\fcolorbox` is modified to accept a background color of none.

(\fcolorbox is particular about its optional arguments, thus the elaborate combinations of \ifthenelse.)

```
7378 \newsavebox{\LWR@colorminipagebox}
7379
7380 \DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
7381 \LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a \fcolorbox:

```
7382 \begin{lrbox}{\LWR@colorminipagebox}%
7383 #5%
7384 \end{lrbox}%
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a \fcolorbox.

```
7385 \ifthenelse{\equal{#4}{none}}%
7386 {% #4 none
7387   \LWR@traceinfo{background is none}%
7388   {% scope the \colorlet
7389     \colorlet{\LWR@currentcolor}{.}%
7390     \color{#2}%
7391     \fbox{%
7392       \color{\LWR@currentcolor}%
7393       \usebox{\LWR@colorminipagebox}%
7394     }% fbox
7395   }% colorlet
7396 }% #4 none
7397 {% #4 not none
7398 \LWR@traceinfo{background not none}%
7399 \IfValueTF{#1}%
7400 {%
7401   \IfValueTF{#3}%
7402   {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7403   {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
7404 }%
7405 {% no value #1
7406   \IfValueTF{#3}%
7407   {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7408   {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
7409 }% no value #1
7410 }% #4 not none
7411 \LWR@traceinfo{\LWRprint@fcolorbox done}%
7412 }
```

\fcolorboxBlock [*{framemodel}*] [*{framecolor}*] [*{boxmodel}*] [*{boxcolor}*] [*{text}*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
7413 \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:

```
7414 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
7415 {%
7416 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
7417 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
7418 \IfValueTF{#7}%
7419 {\begin{minipage}[#5][#6][#7]{#8}}%
7420 {\begin{minipage}[#5][#6][#5]{#8}}%
7421 }%
7422 {%
7423 \end{minipage}%
7424 \end{lrbox}%
7425 \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`.

```
7426 \ifthenelse{\equal{#4}{none}}%
7427 {% #4 none
7428   {% scope the \colorlet
7429     \colorlet{LWR@currentcolor}{.}%
7430     \color{#2}%
7431     \fbox{%
7432       \color{LWR@currentcolor}%
7433       \usebox{\LWR@colorminipagebox}%
7434     }% fbox
7435   }% colorlet
7436 }% #4 none
7437 {% #4 not none
7438   \IfValueTF{#1}%
7439   {%
7440     \IfValueTF{#3}%
7441     {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7442     {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
```

```

7443 }%
7444 {% no value #1
7445 \IfValueTF{#3}%
7446 {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
7447 {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
7448 }% no value #1
7449 }% #4 not none
7450 \LWR@traceinfo{*** finished end fcolorminipage}%
7451 }

```

`\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`.

```

7452 \newcommand*\LWR@restoreorigprintxcolor}{%
7453 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
7454 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
7455 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
7456 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
7457 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
7458 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
7459 }
7460
7461 \appto{\LWR@restoreorigformatting}{%
7462 \LWR@restoreorigprintxcolor%
7463 }

```

If print mode, immediately activate the print-mode enhancements for xcolor:

```

7464 \warpprintonly{\LWR@restoreorigprintxcolor}
7465
7466 \LWR@traceinfo{xcolor patches done}
7467 }{}% xcolor loaded
7468 }% AtBeginDocument

7469 \end{warppall}

```

74 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 402 to redefine the message which is printed for page number references.

loading order `cleveref` and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires `cleveref`. The user’s document may not require `cleveref`, thus the user may never explicitly load it, so during HTML output `lwarp` loads it last. If the user’s document preamble uses `cleveref` options, or functions such as `\crefname`, then `cleveref` may be loaded in the user’s preamble near the end, and `lwarp`’s additional loading of `cleveref` will have no effect.

Table 10 on page 340 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: 7470 `\begin{warpHTML}`

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
7471 \AtEndPreamble{
7472 \RequirePackage{cleveref}
7473 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
7474 \AfterEndPreamble{
7475 \LWR@traceinfo{Patching cleveref.}
```

```
\@@setcref  <{<kindofref>} <{<label>}>
```

```
7476 \renewcommand*{\@@setcref}[2]{#1{\ref{#2}}{-}{-}}
```

```
\@@setcrefrange  <{<text>} <{<label>} <{<label>}>
```

```
7477 \renewcommand{\@@setcrefrange}[3]{%
7478 #1{\ref{#2}}{\ref{#3}}{-}{-}{-}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
7479 \newcommand*\cpagerefFor{for}
```

`\@@setcpageref` $\langle\textit{typeofref}\rangle$ $\langle\textit{label}\rangle$, where *typeofref* is “page” or “pages”

```
7480 \renewcommand*\@@setcpageref[2]{%
7481 #1{\cpagerefFor\ \cref{#2}}-{}-%
7482 }
```

```
7483 \renewcommand*\@@setcpagerefrange[3]{%
7484 #1{\cpagerefFor\ \cref{#2}}-\cref{#3}}-{}-{}-{}-%
7485 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```
7486 \LetLtxMacro\LWR@origlabel\label
7487 % \LetLtxMacro\label\LWR@newlabel
7488 \RenewDocumentCommand{\label}{}{\LWR@newlabel}
7489
7490 \LetLtxMacro\LWR@origref\ref
7491 % \LetLtxMacro\ref\LWR@newref
7492 \RenewDocumentCommand{\ref}{}{\LWR@newref}%
7493
7494 \LetLtxMacro\LWR@origpageref\pageref
7495 % \LetLtxMacro\pageref\LWR@newpageref
7496 \RenewDocumentCommand{\pageref}{}{\LWR@newpageref}
7497
7498 \end{warpHTML}
```

75 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: `7499 \begin{warpHTML}`

Env `picture`

```
7500 \BeforeBeginEnvironment{picture}{\begin{lateximage}}
7501
7502 \AfterEndEnvironment{picture}{\end{lateximage}}

7503 \end{warpHTML}
```

76 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

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

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

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

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

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

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

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

for HTML output: `7504 \begin{warpHTML}`

76.1 Counters and lengths

Ctrl `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
7505 \newcounter{LWR@minipagedepth}
7506 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
7507 \newlength{\LWR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
7508 \newlength{\LWR@minipageheight}
```

76.2 Footnote handling

Also see section 48 for other forms of footnotes. Minipage footnotes are gathered in section 48.5, and then placed into the document in section 76.3.

76.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the \TeX definition and modified to create a `<div>` of class `mpfootnotes`:

```
7509 \def\LWR@endminipage{%
7510   \par
7511   \unskip
7512   \ifvoid\@mpfootins\else
7513     \vskip\skip\@mpfootins
7514     \normalcolor
7515     \LWR@htmldivclass{mpfootnotes}
7516     \LWR@origmedskip
7517     \unvbox\@mpfootins
7518     \LWR@htmldivclassend{mpfootnotes}
7519   \fi
7520   \@minipagefalse
7521   \color@endgroup
7522   \egroup
7523   \expandafter\@iiparbox\@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.

```
7524 \newcommand*\LWR@subminipage}{%
7525 \LWR@stoppars
7526 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
7527 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
7528 \LWR@startpars%
7529 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
7530 \newcommand*{\LWR@endsubminipage}{%
7531 \LWR@stoppars%
7532 \LWR@endminipage% The following empty line is required:
7533
7534 }
```

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
7535 \newbool{LWR@minipagefullwidth}
7536 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output: `7537 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}`
`7538 \end{warpHTML}`

for PRINT output: `7539 \begin{warpprint}`
`7540 \newcommand*{\minipagefullwidth}{}%`
`7541 \end{warpprint}`

for HTML output: `7542 \begin{warpHTML}`

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
7543 \newbool{LWR@minipagethispar}
7544 \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.

```
7545 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
7546 {%
```

Units for printing dimensions to HTML:

```
7547 \uselengthunit{PT}%
```

Compute width, adjusted for frames:

```
7548 \setlength{\LWR@minipagewidth}{#4}%
7549 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
7550   \addtolength{\LWR@minipagewidth}{3em}% room for frames
7551   \setlength{\linewidth}{6in}%
7552   \setlength{\textwidth}{6in}%
7553   \setlength{\textheight}{9in}%
7554 }{%
7555 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
```

Compute height:

```
7556 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
7557 \IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

Track nesting depth:

```
7558 \addtocounter{LWR@minipagedepth}{1}%
```

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.

```
7559 \ifbool{FormatWP}{\newline}{}%
7560 \LWR@stoppars%
```

If FormatWP, add a text frame:

```
7561 \ifbool{FormatWP}{%
7562
7563 \addtocounter{LWR@thisautoid}{1}%
7564 \booltrue{LWR@freezethisautoid}%
7565 \LWR@htmltag{div id="autoid-\arabic{LWR@thisautoid}" class="wpmminipage"}%
7566
7567 }{%
```

Create the <div> tag with optional alignment style:

```
7568 \LWR@traceinfo{minipage: creating div class}%
7569 \LWR@orignewpage%
7570 \LWR@htmltag{div class="minipage" style="%
7571 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{%
7572 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{%
```

```

7573 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
7574 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
7575 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
7576 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
7577 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%

```

Print the width and optional height styles:

```

7578 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
7579 \uselengthunit{PT}%
7580 \ifbool{\LWR@minipagefullwidth}%
7581 {\boolfalse{\LWR@minipagefullwidth}}%
7582 {%
7583   \ifthenelse{\lengthtest{#4}=\linewidth}%
7584   {}%
7585   {width:\rndprintlength{\LWR@minipagewidth} ; }%
7586 }%
7587 \LWR@traceinfo{minipage: about to print the height}%
7588 \IfValueT{#2}{height:\rndprintlength{\LWR@minipageheight} ; }%
7589 "{}}%

```

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.

```

7590
7591 \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.

```

7592 \setlength{\linewidth}{#4}% the original width
7593 \setlength{\textwidth}{6in}%
7594 \setlength{\textheight}{9in}%

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```

7595 \LWR@origraggedright%

```

Resume paragraph tag handling for the contents of the minipage:

```

7596 \LWR@startpars%
7597 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
7598
7599 === begin minipage ===
7600
7601 }{}%

```

```
7602 \LWR@traceinfo{minipage: finished starting the minipage}%
7603 }
```

End the environment with \LaTeX processing and closing tag:

```
7604 {%
7605 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
7606
7607 === end minipage ===
7608
7609 }{%
7610 \LWR@stoppars%
7611 \LWR@endminipage% The following empty line is required:
7612
7613 \ifbool{FormatWP}{%
7614
7615 \LWR@htmlelementend{div}%
7616 \boolfalse{LWR@freezethisautoid}%
7617
7618 }{%
7619 \LWR@htmldivclassend{minipage}%
7620
7621 \LWR@origvspace{1\baselineskip}% required for subcaption
7622 \addtocounter{LWR@minipagedepth}{-1}%
7623 \LWR@startpars%
7624 \ifbool{FormatWP}{\newline}{%}
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
7625 \global\booltrue{LWR@minipagethispar}%
7626 }
```

76.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output:
 \backslash parbox [*pos*] [*height*] [*inner-pos*] {*width*} {*text*}

A parbox uses the minipage code:

```
7627 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
7628 {
7629 \LWR@traceinfo{parbox of width #4}%
7630 \begin{minipage}[#1][#2][#3]{#4}
7631 #5
7632 \end{minipage}
7633 }
```

`\makebox` ($\langle\langle\rangle\rangle$ *posn*) [$\langle\langle width\rangle\rangle$] [$\langle\langle pos\rangle\rangle$] { $\langle\langle text\rangle\rangle$ }

```
7634 \LetLtxMacro\LWR@origmakebox\makebox
7635
7636 \RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
7637 \IfValueTF{#2}%
7638 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```
7639   {% scope
7640   \def\LWR@align{center}%
7641   \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
7642   \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
7643   \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the width argument:

```
7644   \setlength{\LWR@tempwidth}{#2}%
```

`inline-block` allows width and text-align to be used in a ``.

```
7645   \uselengthunit{PT}%
7646   \InlineClass[%
7647     display:inline-block ; %
7648     text-align:\LWR@align\ ; %
7649     width:\rndprintlength{\LWR@tempwidth}%
7650   ]%
7651   {makebox}%
```

Without a width argument, the text is simply used inline:

```
7652   {\mbox{#4}}%
7653   }% scope
7654 }%
7655 {\mbox{#4}}%
7656 }
```

`\framebox` [$\langle\langle width\rangle\rangle$] [$\langle\langle pos\rangle\rangle$] { $\langle\langle text\rangle\rangle$ }

```
7657 \LetLtxMacro\LWR@origframebox\framebox
7658
7659 \RenewDocumentCommand{\framebox}{o o m}{%
```

```
7660 \fbox{\makebox[#1][#2]{#3}}%
7661 }
```

`\LWR@forceminwidth` $\langle \text{length} \rangle$

Sets `\LWR@atleastonept` to be at least 1pt.

```
7662 \newlength{\LWR@atleastonept}
7663
7664 \newcommand*\LWR@forceminwidth[1]{%
7665 \setlength{\LWR@atleastonept}{#1}%
7666 \ifthenelse{%
7667     \lengthtest{\LWR@atleastonept>0pt}\AND%
7668     \lengthtest{\LWR@atleastonept<1pt}}%
7669 }%
7670 {\setlength{\LWR@atleastonept}{1pt}}%
7671 {}%
7672 }
```

`\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.

```
7673 \newcommand*\LWR@blackborderpadding{%
7674 \uselengthunit{PT}%
7675 border:\rndprintlength{\LWR@atleastonept} solid black ; %
7676 padding:\rndprintlength{\fboxsep}%
7677 }
```

`\fbox` $\langle \text{text} \rangle$

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
7678 \let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after `xcolor` may have loaded:

```
7679 \newcommand{\LWRhtml@fbox}[1]{%
7680 \LWR@traceinfo{HTML fbox}
7681 \LWR@forceminwidth{\fboxrule}%
7682 \InlineClass[%
7683 \LWR@blackborderpadding%
7684 ]{fbox}{#1}
7685 }
```

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:

```
7686 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` `{<text>}` Creates a framed HTML <div> of the text.

A print-output version is also supplied below.

```
7687 \newcommand{\fboxBlock}[1]{%
7688 \LWR@forceminwidth{\fboxrule}%
7689 \begin{BlockClass}[%
7690 \LWR@blackborderpadding%
7691 ]{\fboxBlock}
7692 #1
7693 \end{BlockClass}
7694 }
```

Env `fminipage` [`<align>`] [`<height>`] [`<align>`] [`<width>`]

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```
7695 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
7696 {%
7697 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
7698 \LWR@forceminwidth{\fboxrule}%
7699 \setlength{\LWR@tempwidth}{#4}%
7700 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
7701 \begin{BlockClass}[%
7702 \LWR@blackborderpadding ; %
7703 \uselengthunit{PT}%
7704 \IfValueT{#2}{height:\rndprintlength{\LWR@tempheight} ; }%
7705 width:\rndprintlength{\LWR@tempwidth}%
7706 ]{fminipage}%
7707 }
7708 {%
7709 \end{BlockClass}%
7710 \LWR@traceinfo{fminipage done}%
7711 }
```

`\raisebox` [`<raiselen>`] [`<height>`] [`<depth>`] [`<text>`]

```
7712 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
7713
7714 \RenewDocumentCommand{\raisebox}{m o o m}{%
```

```
7715 #4%
7716 }
```

```
7717 \end{warpHTML}
```

for HTML & PRINT: 7718 \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.

```
7719 \newsavebox{\LWR@fminipagebox}
7720
7721 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
7722 {%
```

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:

```
7723 \IfValueTF{#3}%
7724 {\def\LWR@thisalign{#3}}
7725 {\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.

```
7726 \IfValueTF{#2}%
7727 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
7728 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
7729 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
7730 \IfValueTF{#2}%
7731 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
7732 {\minipage[#1]{#4}}%
7733 }
7734 {%
```

Close the inner minipage and the LR box with the contents:

```
7735 \endminipage%
7736 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
7737 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
7738 \endminipage%
7739 }
```

```
7740 \end{warppall}
```

for PRINT output: 7741 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock  {\langle text \rangle}
```

Creates a framed HTML <div> around the text.

```
7742 \let\fboxBlock\fbox
```

```
Env  fminipage  [\langle align \rangle] [\langle height \rangle] [\langle align \rangle] {\langle width \rangle}
```

Creates a frame around its contents.

```
7743 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}
7744 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}
```

```
7745 \end{warpprint}
```

77 Direct formatting

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

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

For high-level block and inline custom CSS classes, see section [42.8](#).

for HTML output: 7746 `\begin{warpHTML}`

`\LWR@HTMLtextstyle` $\langle FormatWP style \rangle$ $\langle class \rangle$ $\langle text \rangle$

If `FormatWP`, adds an explicit style to the text span class. This is used by LibreOffice to mark its imported text using the given style.

```
7747 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
7748 \ifbool{FormatWP}%
7749 {\LWR@htmlspanclass[#1]{#2}{#3}}%
7750 {\LWR@htmlspanclass{#2}{#3}}%
7751 }
```

`\emph` $\langle text \rangle$

```
7752 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
7753 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
7754 \LetLtxMacro{\emph}{\LWR@HTMLemph}
```

`\textmd` $\langle text \rangle$

```
7755 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
7756 \LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}}%
7757 }
7758 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
7759 }
7760 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

`\textbf` $\langle text \rangle$

```
7761 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
7762 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
7763 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm` $\langle text \rangle$

```
7764 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
7765 \LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}}%
7766 }
7767 }
7768 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
7769 }
7770 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf` $\langle text \rangle$

```
7771 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
7772 \LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
7773 }
7774 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
7775 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt` $\langle text \rangle$

```
7776 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
7777 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
7778 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup` $\langle text \rangle$

```
7779 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
7780 \LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
7781 }
7782
7783 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
7784
7785 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit` $\langle text \rangle$

```
7786 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
7787 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7788 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc` $\langle text \rangle$

```
7789 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
7790 \LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
7791 }
7792
7793 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7794
7795 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

`\textsl` $\langle text \rangle$

```
7796 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
7797 \LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
7798 }
7799
```

```
7800 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7801
7802 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

`\textnormal` $\langle text \rangle$

```
7803 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7804 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7805 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}
```

```
7806 \DeclareRobustCommand{\LWR@nullrmfamily}{}
7807 \DeclareRobustCommand{\LWR@nullsffamily}{}
7808 \DeclareRobustCommand{\LWR@nullttfamily}{}
7809 \DeclareRobustCommand{\LWR@nullbfseries}{}
7810 \DeclareRobustCommand{\LWR@nullmdseries}{}
7811 \DeclareRobustCommand{\LWR@nullupshape}{}
7812 \DeclareRobustCommand{\LWR@nullslshape}{}
7813 \DeclareRobustCommand{\LWR@nullscshape}{}
7814 \DeclareRobustCommand{\LWR@nullitshape}{}
7815 \DeclareRobustCommand{\LWR@nullem}[1]{}
7816 \DeclareRobustCommand{\LWR@nullnormalfont}{}

```

`\LWR@nullfonts` Removes formatting during filename operations.

```
7817 \newcommand*{\LWR@nullfonts}{%
7818 \LetLtxMacro{\emph}{\LWR@nullemph}%
7819 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7820 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7821 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7822 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
7823 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7824 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7825 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7826 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7827 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7828 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
7829 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7830 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7831 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7832 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7833 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7834 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7835 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7836 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7837 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7838 \LetLtxMacro{\em}{\LWR@nullem}%
7839 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7840 \renewcommand*{\HTMLUnicode}[1]{%

```

```
7841 \renewcommand*{\HTMLentity}[1]{}%
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
7842 \renewcommand*{\&}{and}%
7843 \renewcommand{\textsuperscript}[1]{##1}%
7844 \renewcommand{\textsubscript}[1]{##1}%
7845 \LetLtxMacro\underline\LWR@origunderline%
7846 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7847 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
7848 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%
7849 \DeclareRobustCommand{\LWR@subsingledollar}[1]{}%
7850 }
```

`\mdseries`

```
7851 \renewcommand*{\mdseries}{}%
```

`\bfseries`

```
7852 \renewcommand*{\bfseries}{}%
```

`\rmfamily`

```
7853 \renewcommand*{\rmfamily}{}%
```

`\sffamily`

```
7854 \renewcommand*{\sffamily}{}%
```

`\ttfamily`

```
7855 \renewcommand*{\ttfamily}{}%
```

`\upshape`

```
7856 \renewcommand*{\upshape}{}%
```

`\itshape`

```
7857 \renewcommand*{\itshape}{}%
```

`\scshape`

```
7858 \renewcommand*\scshape{}
```

`\normalfont`

```
7859 \renewcommand*\normalfont{}
```

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

```
7860 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}}
```

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

```
7861 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}}
```

`\textsuperscript` $\langle text \rangle$

```
7862 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\@textsuperscript` $\langle text \rangle$

```
7863 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript` $\langle text \rangle$

```
7864 \AtBeginDocument{
7865 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
7866 }
```

`\@textsubscript` $\langle text \rangle$

```
7867 \AtBeginDocument{
7868 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
7869 }
```

`\up` $\langle text \rangle$ Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7870 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup` $\langle text \rangle$ Prints superscript.

Supports `fntcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7871 \AtBeginDocument{\let\fup\textsuperscript}
```

`\underline` $\langle text \rangle$

```
7872 \renewcommand{\underline}[1]{%
7873 \LWR@HTMLtextstyle%
7874   {text-decoration:underline;text-decoration-skip}%
7875   {\underline}{#1}%
7876 }
```

`\hfill`

```
7877 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
7878 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
7879 \renewcommand*{\dotfill}{\dots}
```

```
7880 \end{warpHTML}
```

78 Skips, spaces, font sizes

for HTML output: 7881 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
7882 \let\LWR@origcomma\,
7883 \let\LWR@origtilde~
7884 \let\LWR@origenskip\enskip
7885 \let\LWR@origquad\quad
7886 \let\LWR@origqqquad\qqquad
7887 \let\LWR@origspace\hspace
```

```

7888 \let\LWR@origvspace\vspace
7889 \let\LWR@origrule\rule
7890 \let\LWR@origmedskip\medskip
7891 \let\LWR@origtextellipsis\textellipsis

```

Direct-formatting space commands become HTML entities:

```

7892 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space

7893
7894 \renewcommand*{~}{\HTMLentity{nbsp}}
7895
7896 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}

```

Direct-formatting font sizes are ignored:

```

7897 \let\LWR@orignormalsize\normalsize
7898 \let\LWR@origsmall\small
7899 \let\LWR@origfootnotesize\footnotesize
7900 \let\LWR@origscriptsize\scriptsize
7901 \let\LWR@origtiny\tiny
7902 \let\LWR@origlarge\large
7903 \let\LWR@origLarge\Large
7904 \let\LWR@origLARGE\LARGE
7905 \let\LWR@orighuge\huge
7906 \let\LWR@origHuge\Huge
7907 \renewcommand*{\normalsize}{}
7908 \renewcommand*{\small}{}
7909 \renewcommand*{\footnotesize}{}
7910 \renewcommand*{\scriptsize}{}
7911 \renewcommand*{\tiny}{}
7912 \renewcommand*{\large}{}
7913 \renewcommand*{\Large}{}
7914 \renewcommand*{\LARGE}{}
7915 \renewcommand*{\huge}{}
7916 \renewcommand*{\Huge}{}
7917
7918 \renewcommand*{\onecolumn}{}
7919
7920 \renewcommand{\twocolumn}[1][1]{}
7921
7922 #1
7923
7924 }

```

`\newline` Uses the HTML `
` element.

```

7925 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7926 \let\newline\LWR@newlinebr

```

`\` Redefined to `\LWR@endofline` or `\LWR@tabularendofline`.

`\LWR@endofline` * [*len*]

`\` is assigned to `\LWR@endofline` at `\LWR@LwarpStart`.

Inside `tabular`, `\` is temporarily changed to `\LWR@tabularendofline`.

```
7927 \LetLtxMacro\LWR@origendofline\
7928 \NewDocumentCommand{\LWR@endofline}{s o}
7929 {%
7930 \newline%
7931 }
```

`\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.

`\hspace`
`\enskip`
`\quad`
`\qquad`

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.

```
7932 \newcommand*{\LWR@minipagestartpars}{%
7933 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
7934 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
7935 \newcommand*{\LWR@minipagestoppars}{%
7936 \ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
7937 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
7938 \renewcommand*{\quad}{%
```

```

7939 \LWR@minipagestoppars%
7940 \HTMLUnicode{2001}%
7941 \LWR@minipagestartpars%
7942 }

```

`\qqad` Handles special minipage & horizontal space interactions.

```
7943 \renewcommand*{\qqad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```

7944 \renewcommand*{\enskip}{%
7945 \LWR@minipagestoppars%
7946 \HTMLUnicode{2000}%
7947 \LWR@minipagestartpars%
7948 }

```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

```

Len \WR@tempheight 7949 \newlength{\LWR@tempwidth}
Len \WR@tempraise 7950 \newlength{\LWR@tempheight}
7951 \newlength{\LWR@tempraise}

```

`\LWR@hspace` * $\langle length \rangle$

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 `\qqad`.

```

7952 \NewDocumentCommand{\LWR@hspace}{s m}{%
7953 \setlength{\LWR@tempwidth}{#2}%

```

If `\fill`, change to `\qqad`:

```

7954 \ifnum\gluestretchorder\LWR@tempwidth>0%
7955 \setlength{\LWR@tempwidth}{2em}%
7956 \fi%

```

Only if the width is not zero:

```
7957 \ifdimcomp{\LWR@tempwidth}{=}{0pt}{-}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7958 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
7959 \ifdimcomp{\LWR@tempwidth}{=}{.16667em}%
7960 {%
7961 \HTMLUnicode{2009}% thin breakable space
7962 }%
```

Print the span with the converted width. Not rounded.

```
7963 {%
7964 \uselengthunit{PT}%
7965 \LWR@htmltagc{%
7966 span style="width:\printlength{\LWR@tempwidth}; %
7967 display:inline-block"%
7968 }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```
7969 \ifbool{FormatWP}{%
7970 \setlength{\LWR@templengthone}{\LWR@tempwidth}%
7971 \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
7972 \quad%
7973 \addtolength{\LWR@templengthone}{-1em}%
7974 }%
7975 }{ }%
```

Close the span:

```
7976 \LWR@htmltagc{/span}%
7977 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7978 \LWR@minipagestartpars%
7979 }% width not 0
7980 }
```

```
\LWR@nohspace * {<length>}
```

Used to disable \hspace while creating description \items.

```
7981 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

```
\hspace * {<length>}
```

Handles special minipage & horizontal space interactions.

```
7982 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` * $\langle length \rangle$ Nullified vspace.

```
7983 \NewDocumentCommand{\LWR@vspace}{s m}{}

```

`\vspace` * $\langle length \rangle$ Nullified.

```
7984 \let\vspace\LWR@vspace

```

`\linebreak` [$\langle num \rangle$] Inserts an HTML br tag.

```
7985 \renewcommand*{\linebreak}[1] [] {\newline}

```

`\nolinebreak` [$\langle num \rangle$]

```
7986 \renewcommand*{\nolinebreak}[1] [] {}

```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
7987 \renewcommand*{\pagebreak}[1] [] {
7988
7989 }

```

`\nopagebreak` [$\langle num \rangle$]

```
7990 \renewcommand*{\nopagebreak}[1] [] {}

```

`\enlargethispage` * $\langle len \rangle$

```
7991 \RenewDocumentCommand{\enlargethispage}{s m}{}

```

`\clearpage`
`\cleardoublepage`

```
7992 \renewcommand*{\clearpage}{}
7993 \renewcommand*{\cleardoublepage}{}

```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
7994 \newcommand*{\LWR@currenttextcolor}{black}

```

```
\LWR@rule  [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
7995 \NewDocumentCommand{\LWR@rule}{o m m}{%
```

The width is copied into a temporary \TeX length, from which comparisons and conversions may be made:

```
7996 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```
7997 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
7998 {}% zero- width
7999 {% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
8000 \ifthenelse{%
8001     \lengthtest{\LWR@tempwidth>0pt}\AND%
8002     \lengthtest{\LWR@tempwidth<1pt}}%
8003 }%
8004 {\setlength{\LWR@tempwidth}{1pt}}{}}%
```

Likewise with height:

```
8005 \setlength{\LWR@tempheight}{#3}%
8006 \ifthenelse{%
8007     \lengthtest{\LWR@tempheight>0pt}\AND%
8008     \lengthtest{\LWR@tempheight<1pt}}%
8009 }%
8010 {\setlength{\LWR@tempheight}{1pt}}{}}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
8011 \LWR@minipagestoppars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in \TeX code.

```
8012 \uselengthunit{PT}%
```

```

8013   \LWR@htmltagc{%
8014   span
8015   style="%

```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```

8016   \ifbool{FormatWP}{}{background:\LWR@currenttextcolor ; }%

```

The width and height are printed, converted to PT:

```

8017   width:\printlength{\LWR@tempwidth} ; %
8018   height:\printlength{\LWR@tempheight} ; %

```

The raise height is converted to a css transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a \TeX length allows a typical \TeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \TeX length limits the allowable syntax. To do: A superior method would compute a ratio of \TeX ex height, then print that to HTML with an ex unit.

```

8019   \IfValueT{#1}%
8020   {%
8021     \setlength{\LWR@tempraise}{Opt-#1}%
8022     \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
8023     \LWR@orignewline%
8024     -ms-transform: translate(0pt,\printlength{\LWR@tempraise}); %
8025     \LWR@orignewline%
8026     -webkit-transform: translate(0pt,\printlength{\LWR@tempraise}); %
8027     \LWR@orignewline%
8028     transform: translate(0pt,\printlength{\LWR@tempraise}); %
8029     \LWR@orignewline%
8030   }%

```

Display inline-block to place the span inline with the text:

```

8031   display:inline-block;"%
8032   }%

```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```

8033   \ifbool{FormatWP}{%
8034     \setlength{\LWR@templengthone}{\LWR@tempwidth}%
8035     \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
8036       \_}%
8037     \addtolength{\LWR@templengthone}{-1em}%
8038   }%

```

```
8039     }{ }%
```

Close the span:

```
8040     \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
8041     \LWR@minipagestartpars%
8042 }% non-zero width
8043 }
```

```
\rule  [<raise>] {<width>} {<height>}
```

Handles special minipage & horizontal space interactions.

```
8044 \renewcommand{\rule}{\LWR@rule}
```

```
8045 \end{warpHTML}
```

79 \phantomsection

for HTML output: 8046 \begin{warpHTML}

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into table of contents:

```
8047 \newcommand*{\phantomsection}{%
8048 \section*{}%
8049 }
```

```
8050 \end{warpHTML}
```

80 \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>

80.1 HTML logos

for HTML output: 8051 `\begin{warpHTML}`

`\TeX` \TeX

`latexlogo` is a CSS class used to properly typeset the E and A in \TeX and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in \TeX , Lua \TeX , Con \TeX t, etc.

```
8052 \let\LWR@origTeX\TeX
8053
8054 \newcommand*{\LWR@TeX}
8055 {%
8056   \InlineClass{latexlogofont}%
8057   {%
8058     \LWR@HTMLtextstyle%
8059     {text-transform:uppercase}%
8060     {latexlogo}%
8061     {T\textsubscript{e}X}%
8062   }%
8063 }
```

`\LaTeX` \TeX , $\TeX 2\epsilon$
`\LaTeXe`

```
8064 \let\LWR@origLaTeX\LaTeX
8065
8066 \newcommand*{\LWR@LaTeX}
8067 {%
8068   \InlineClass{latexlogofont}%
8069   {%
8070     \LWR@HTMLtextstyle%
8071     {text-transform:uppercase}%
8072     {latexlogo}%
8073     {L\textsuperscript{a}T\textsubscript{e}X}%
8074   }%
8075 }
8076
8077 \let\LWR@origLaTeXe\LaTeXe
8078
8079 \renewcommand*{\LaTeXe}
```

```
8080 {\LaTeX\InlineClass{latexlogofont}%
8081 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

`\LuaTeX` `LualTeX`, `LualTeX`
`\LuaLaTeX`

```
8082 \newcommand*\LWR@LuaTeX{\InlineClass{latexlogofont}{Lua}\TeX}
8083 \newcommand*\LWR@LuaLaTeX{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

`\XeTeX` `XeTeX`, `XeTeX`
`\XeLaTeX`

`xetexlogo` is a css class which aligns the backwards E in `XeTeX` and spaces `TeX` appropriately.

`xelatexlogo` is a css class which aligns the backwards E in `XeTeX` and spaces `TeX` appropriately.

```
8084 \newcommand*\Xe}
8085 {X\textsubscript{\HTMLUnicode{18e}}}
8086 \newcommand*\LWR@XeTeX{\InlineClass{xetexlogo}{\Xe}\TeX}
8087 \newcommand*\LWR@XeLaTeX{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

`\ConTeXt` `ConTeXt`

```
8088 \newcommand*\LWR@ConTeXt}
8089 {\InlineClass{latexlogofont}{Con}\TeX{}}%
8090 \InlineClass{latexlogofont}{t}}
```

`\BibTeX` `BibTeX`, `MakeIndex`
`\MakeIndex`

```
8091 \providecommand*\BibTeX}
8092 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
8093
8094 \newcommand*\MakeIndex}
8095 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}
```

`\AmS` `AMS`

`amslogo` is a css class used for the `AMS` logo.

```
8096 \AtBeginDocument{\DeclareDocumentCommand\AmS}{}
8097 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}
```

`\MiKTeX` `MiKTeX`

```
8098 \newcommand*\MiKTeX{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX LyX`

`lyxlogo` is a CSS class used for the LyX logo.

```
8099 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
8100 \end{warpHTML}
```

80.2 Print logos

for PRINT output:

```
8101 \begin{warpprint}
8102 \newcommand*{\XeTeXrevE}
8103   {\hspace{- .1667em}\raisebox{- .5ex}{\reflectbox{E}}\hspace{- .125em}}
8104 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
8105 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
8106 \providecommand*{\AMS}{%
8107 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
8108 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
8109 \newcommand*{\LyX}{\textsf{LyX}}
8110 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
8111 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
8112 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
8113 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
8114 \providecommand*{\ConTeXt}{\mbox{Con\TeX{t}}}
8115 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
8116 \end{warpprint}
```

81 \AtBeginDocument, \AtEndDocument

for HTML output: `8117 \begin{warpHTML}`

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.

`\LWR@LwarpEnd`

```
8118 \AfterEndPreamble{\LWR@LwarpStart}
8119 \AtEndDocument{\LWR@LwarpEnd}

8120 \end{warpHTML}
```

82 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.

The following adjustments apply to the lwarp-* package listings:

File 2 `lwarp-patch-komascript.sty`

§ 83 Package **patch-komascript**

Pkg `lwarp-patch-komascript` Patches for komascript classes.

lwarp loads this package when `scrbook`, `scrartcl`, or `scrreprt` are detected.

Many features are ignored during the HTML conversion. The goal is source-level compatibility.

`\titlehead`, `\subject`, `\captionformat`, `\figureformat`, and `\tableformat` are not yet emulated.

 **Not yet tested!** [Please send bug reports!](#)

Some features have not yet been tested. Please contact the author with any bug reports.

for HTML output: `1 \ProvidesPackage{lwarp-patch-komascript}`

`typearea` is emulated.

`2 \RequirePackage{lwarp-typearea}`

`tocbasic` is emulated.

`3 \RequirePackage{lwarp-tocbasic}`

`scrxextend` patches most of the new macros.

`4 \RequirePackage{lwarp-scrxextend}`

The `\minisec` is placed inside a `<div>` of class `minisec`.

```
5 \renewcommand*{\minisec}[1]{
6 \begin{BlockClass}{minisec}
7 #1
8 \end{BlockClass}
9 }
```

The part and chapter preambles are placed as plain text just after each heading.

```
10 \@ifundefined{setpartpreamble}{}{
```

```

11 \RenewDocumentCommand{\setpartpreamble}{o o +m}{%
12 \renewcommand{\part@preamble}{#3}%
13 }
14 }
15
16 \@ifundefined{setchapterpreamble}{}{
17 \RenewDocumentCommand{\setchapterpreamble}{o o +m}{%
18 \renewcommand{\chapter@preamble}{#3}%
19 }
20 }

```

Simple captions are used in all cases.

```

21 \LetLtxMacro\captionbelow\caption
22 \LetLtxMacro\captionabove\caption
23
24 \LetLtxMacro\captionofbelow\captionof
25 \LetLtxMacro\captionofabove\captionof
26
27 \RenewDocumentEnvironment{captionbeside}{o m o o o s}
28 {}
29 {%
30 \IfValueTF{#1}%
31 {\caption[#1]{#2}}%
32 {\caption{#2}}%
33 }
34
35 \RenewDocumentEnvironment{captionofbeside}{m o m o o o s}
36 {}
37 {%
38 \IfValueTF{#2}%
39 {\captionof{#1}[#2]{#3}}%
40 {\captionof{#1}{#3}}%
41 }
42
43 \RenewDocumentCommand{\setcapindent}{s m}{}
44 \renewcommand*\setcaphanging{}
45 \renewcommand*\setcapwidth}[2] [] {}
46 \renewcommand*\setcapdynwidth}[2] [] {}
47 \RenewDocumentCommand{\setcapmargin}{s o m}{}

```

File 3 **lwarp-a4.sty**

§ 84 Package **a4**

Pkg a4 a4 is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{a4}
2 \newcommand*{\WideMargins}{}
```

File 4 **lwarp-a4wide.sty**

§ 85 Package **a4wide**

Pkg a4wide a4wide is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{a4wide}
```

File 5 **lwarp-a5comb.sty**

§ 86 Package **a5comb**

Pkg a5comb a5comb is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{a5comb}
```

File 6 **lwarp-abstract.sty**

§ 87 Package **abstract**

(Emulates or patches code by PETER WILSON.)

Pkg abstract abstract is supported and patched by lwarp.

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

for HTML output: 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*{\abstitleskip}%
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{\abstitleskip}%
35 \fi
36 \fi
37 \vspace{\abstitleskip}%
38 \fi
39 \put@bsintoc%
40 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42 }
43 \else
44 \renewenvironment{abstract}{%
45 \if@bsrunin
46 \else
47 \if@bsstyle
48 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49 \else
50 \ifnumber@bs
51 \num@bs
52 \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%

```

```

56 %%          \vspace{\abstitleskip}%
57           \fi
58           \fi
59           \vspace{\abstitleskip}%
60           \fi
61           \put@bsintoc%
62           \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63           {\par\end{@bstr@ctlist}}
64 \fi
65

```

File 7 **lwarp-adjmulticol.sty**

§ 88 Package **adjmulticol**

(Emulates or patches code by BORIS VEYTSMAN.)

Pkg adjmulticol adjmulticol is emulated.

Emulation similar to multicols is used, with adjusted margins. If the number of columns is specified as 1, it is set so, but if two or greater are used, lwarp allows a variable number of columns up to three.

for HTML output: 1 \LWR@ProvidesPackageDrop{adjmulticol}

2 \RequirePackage{multicol}

adjmulticols * {<numcols>} {<left margi>} {<right margin>}

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@mcolstype}{multicols}%

10 \ifnumcomp{#2}{=}{1}{\renewcommand*{\LWR@mcolstype}{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@mcolstype}%
17 }
18 {\endBlockClass}
```

File 8 lwarp-addlines.sty

§ 89 Package **addlines**

(Emulates or patches code by WILL ROBERTSON.)

Pkg **addlines** addlines is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{addlines}

2 \newcommand*\addlines[1][1]{
3 \let\addline\addlines
4 \newcommand*\removelines[1][1]{
5 \let\removeline\removelines
```

File 9 lwarp-afterpage.sty

§ 90 Package **afterpage**

(Emulates or patches code by DAVID CARLISLE.)

Pkg **afterpage** Emulated.

for HTML output: Discard all options for lwarp-afterpage:

```
1 \LWR@ProvidesPackageDrop{afterpage}

2 \newcommand{\afterpage}[1]{#1}
```

File 10 `lwarp-algorithmicx.sty`§ 91 Package **algorithmicx***(Emulates or patches code by SZÁSZ JÁNOS.)*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 264.1.

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 \ifbool{FormatWP}{%
14 \setlength{\LWR@templengthone}{\the\ALG@thistlm}%
15 \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
16 \quad%
17 \addtolength{\LWR@templengthone}{-1em}%
18 }%
19 }{}%
20 \LWR@htmltagc{/span}%
21 }%
22
23 \let\LWR@origComment\Comment%
24
25 \renewcommand{\Comment}[1]{%
26   \InlineClass{floatright}{\LWR@origComment{#1}}%
27 }%
28 }
29
30 \renewcommand\algorithmiccomment[1]{%
31 \hfill\HTMLunicode{25B7} #1% white right triangle

```

```

32 }%
33 \end{warpHTML}

```

File 11 **lwarp-alltt.sty**

§ 92 Package **alltt**

(Emulates or patches code by JOHANNES BRAAMS.)

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 12 **lwarp-amsthm.sty**

§ 93 Package **amsthm**

(Emulates or patches code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

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}%

```

Table 12: AMSthm package — 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.

```

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 13 **lwarp-anonchap.sty**

§ 94 Package **anonchap**

(Emulates or patches code by PETER WILSON.)

Pkg anonchap anonchap is emulated.

 **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 \TeX 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\@chaptformat##1{%
4     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
5   }%
6 }
7
8 \newcommand{\restorechapter}{%
9 \let\@chaptformat\@seccntformat%
10 }

```

File 14 **lwarp-anysize.sty**

§ 95 Package **anysize**

(Emulates or patches code by MICHAEL SALZENBERG, THOMAS ESSER.)

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 15 **lwarp-appendix.sty**

§ 96 Package **appendix**

(Emulates or patches code by PETER WILSON.)

Pkg appendix appendix is patched for use by lwarp.

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

for HTML output: 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 16 **lwarp-arabicfront.sty**

§ 97 Package **arabicfront**

Pkg arabicfront arabicfront is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{arabicfront}

File 17 **lwarp-atbegshi.sty**§ 98 Package **atbegshi***(Emulates or patches code by HEIKO OBERDIEK.)*

Pkg atbegshi Emulated.

for HTML output: Discard all options for lwarp-atbegshi:

```

1 \LWR@ProvidesPackageDrop{atbegshi}[2011/10/05]

2 \newcommand*\AtBeginShipout}[1]{}
3 \newbox\AtBeginShipoutBox
4 \newcommand*\AtBeginShipoutNext}[1]{}
5 \newcommand*\AtBeginShipoutFirst}[1]{}
6 \newcommand*\AtBeginShipoutDiscard{}
7 \newcommand*\AtBeginShipoutInit{}
8 \newcommand*\AtBeginShipoutAddToBox}[1]{}
9 \newcommand*\AtBeginShipoutAddToBoxForeground}[1]{}
10 \newcommand*\AtBeginShipoutUpperLeft}[1]{}
11 \newcommand*\AtBeginShipoutUpperLeftForeground}[1]{}
12 \newcommand*\AtBeginShipoutOriginalShipout}[1]{}
13 \def\AtBeginShipoutBoxWidth{Opt}
14 \def\AtBeginShipoutBoxHeight{Opt}
15 \def\AtBeginShipoutBoxDepth{Opt}
16

```

File 18 **lwarp-authblk.sty**§ 99 Package **authblk***(Emulates or patches code by PATRICK W. DALY.)*

Pkg authblk authblk is patched for HTML.

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 55.7.

(Emulates or patches 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 19 **lwarp-balance.sty**

§ 100 Package **balance**

(Emulates or patches code by PATRICK W. DALY.)

Pkg balance Emulated.

for HTML output: Discard all options for lwarp-balance:

```

1 \LWR@ProvidesPackageDrop{balance}

2 \newcommand*{\balance}{}
3 \newcommand*{\nobalance}{}

```

File 20 **lwarp-bigdelim.sty**

§ 101 Package **bigdelim**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

`\pkg bigdelim` `bigdelim` is used as-is for print or `lateximage`, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the `\multirow` which indicates the actual height of the delimiter. The delimiter character is given a `` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`

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

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

```

      a  b
left { c  d
      e  f
      g  h

```

for HTML output: First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case `bigdelim` was not loaded:

```
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:delimiter\rangle} {\langle:#rows\rangle} {\langle:width\rangle} [\langle: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 21 **lwarp-bigstrut.sty**

§ 102 Package **bigstrut**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg bigstrut bigstrut is used as-is for print or lateximage, and patched for HTML.

for HTML output:

```
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 22 **lwarp-bookmark.sty**

§ 103 Package **bookmark**

(Emulates or patches code by HEIKO OBERDIEK.)

Pkg bookmark bookmark is emulated.

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 23 **lwarp-booktabs.sty**

§ 104 Package **booktabs**

(Emulates or patches code by SIMON FEAR.)

Pkg booktabs booktabs is emulated during HTML output, and used as-is during print output and inside an HTML lateximage.

for HTML output: 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 24 **lwarp-boxedminipage.sty**

§ 105 Package **boxedminipage**

Pkg boxedminipage boxedminipage is superceded by boxedminipage2e.

for HTML output: 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

File 25 **lwarp-boxedminipage2e.sty**

§ 106 Package **boxedminipage2e**

(Emulates or patches code by SCOTT PAKIN.)

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 26 `lwarp-breakurl.sty`

§ 107 Package **breakurl**

(Emulates or patches code by VILAR CAMARA NETO.)

Pkg `breakurl` `breakurl` is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{breakurl}

2 \LetLtxMacro\burl\url
3
4 \NewDocumentCommand{\LWR@burlaltb}{0{} +m m}{%
5 \LWR@subhyperref{#3}{#2}% order reversed from \href
6 \endgroup%
7 \LWR@ensuredoingapar%
8 }
9
10 \DeclareRobustCommand*\burlalt{%
11 \LWR@ensuredoingapar%
12 \begingroup%
13 \catcode'\_ =12
14 \LWR@burlaltb%
15 }
16 \LetLtxMacro\urlalt\burlalt

```

File 27 `lwarp-cancel.sty`

§ 108 Package **cancel**

Pkg `cancel` `cancel` is used as-is for SVG math, and emulated for HTML text output.

for HTML output: 1 \LWR@ProvidesPackagePass{cancel}

\cancelto is math-only, so is used as-is.

```
2 \LetLtxMacro\LWR@origcancel\cancel
3 \LetLtxMacro\LWR@origbcancel\bcancel
4 \LetLtxMacro\LWR@origxcancel\xcancel
5
6 \appto{\LWR@restoreorigformatting}{%
7 \LetLtxMacro\cancel\LWR@origcancel%
8 \LetLtxMacro\bcancel\LWR@origbcancel%
9 \LetLtxMacro\xcancel\LWR@origxcancel%
10 }
```

\LWR@cancelcolor {<text>} {<color>} {<class>} {<colorstyle>} {<FormatWPstyle>}

Add colors if not empty:

```
11 \newcommand{\LWR@cancelcolor}[5]{%
12 \ifcempty{#2}%
13 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
14 {\LWR@htmlspanclass[#5;#4:\#\LWR@tempcolor]{#3}{#1}}%
15 }
```

\cancel {<text>}

```
16 \DeclareRobustCommand{\cancel}[1]{%
17 \begingroup%
18 \CancelColor%
19 \LWR@findcurrenttextcolor%
20 \color{black}%
21 \LWR@cancelcolor{#1}{\LWR@tempcolor}{sout}{text-decoration-color}%
22 {text-decoration:line-through}%
23 \endgroup%
24 }
25
26 \LetLtxMacro\bcancel\cancel
27 \LetLtxMacro\xcancel\cancel
```

File 28 **lwarp-caption2.sty**

§ 109 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 29 `lwarp-ccaption.sty`

§ 110 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 30 `lwarp-changepage.sty`

§ 111 Package **changepage**

(Emulates or patches code by PETER WILSON.)

Pkg `changepage` `changepage` is emulated.

for HTML output: Discard all options for `lwarp-changepage`:

```
1 \LWR@ProvidesPackageDrop{changepage}

2 \newif\ifoddpge
3 \DeclareRobustCommand{\checkoddpge}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changepage}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}

8 \newcommand*{\strictpagecheck}{}
9 \newcommand*{\easypagecheck}{}

```

File 31 `lwarp-chngpage.sty`

§ 112 Package **chngpage**

Pkg `chngpage` `chngpage` is superseded by `changepage`.

for HTML output: `1 \LWR@loadnever{chngpage}{changepage}`

 File 32 **lwarp-chappg.sty**

 § 113 Package **chappg**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg chappg chappg is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{chappg}

2 \renewcommand{\pagenumbering}[2] [] {}
3 \providecommand{\chappgsep}{--}
```

 File 33 **lwarp-color.sty**

 § 114 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 34 **lwarp-crop.sty**

 § 115 Package **crop**

(Emulates or patches code by MELCHIOR FRANZ.)

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 35 **lwarp-cuted.sty**

 § 116 Package **cuted**

(Emulates or patches code by SIGITAS TOLUŠIS.)

Pkg cuted cuted is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{cuted}

2 \newenvironment{strip}{}{}

3 \newskip\stripsep

4 \def\oldcolsbreak#1{}

 File 36 **lwarp-cutwin.sty**

 § 117 Package **cutwin**

(Emulates or patches code by PETER WILSON AND ALAN HOENIG.)

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 37 **lwarp-dblfnote.sty**

§ 118 Package **dblfnote**

(Emulates or patches code by HIROSHI NAKASHIMA.)

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 38 **lwarp-dcolumn.sty**

§ 119 Package **dcolumn**

Pkg dcolumn dcolumn is emulated by the lwarp core.

1 \LWR@ProvidesPackageDrop{dcolumn}

 File 39 **lwarp-draftwatermark.sty**

 § 120 Package **draftwatermark**

(Emulates or patches code by SERGIO CALLEGARI.)

Pkg draftwatermark draftwatermark is emulated.

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 40 **lwarp-easy-todo.sty**

 § 121 Package **easy-todo**

(Emulates or patches code by JUAN RADA-VILELA.)

Pkg easy-todo easy-todo is patched for use by lwarp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{easy-todo}

```

`\listoftodos` Modified to correct buggy use of `\flushright`.

```

2 \let\LWR@origlistoftodos\listoftodos
3
4 \renewcommand{\listoftodos}{%
5 \begingroup
6 \renewcommand{\flushright}{}
7 \LWR@origlistoftodos
8 \endgroup
9 }

```

`\todoii` Modified to use `\textcolor` instead of `\color`.

```

10 \renewcommand{\todoii}[2]{%
11 \ifthenelse{\equal{\@todoobeyfinal}{true}}{%
12   \ifoptionfinal{\todoenable{false}}{\todoenable{true}}%
13 }{}%
14 \ifthenelse{\equal{\@todoenable}{true}}{%
15 \refstepcounter{todos}%
16 \noindent{%
17   \todocolor%
18   \LWR@textcurrentcolor{%
19     \normalfont\scriptsize{\bfseries{\thetodos.#1}}%
20   }%
21 }%
22 \addcontentsline{lod}{todos}{\protect{\thetodos.#2}}%
23 }{}%
24 }

```

File 41 **lwarp-ebook.sty**

§ 122 Package **ebook**

(Emulates or patches code by JØRGEN STEENSGAARD.)

Pkg ebook ebook is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{ebook}

2 \setcounter{secnumdepth}{0}
3 \setcounter{tocdepth}{2}
4
5 \providecommand{\pagefill}[1][0.001mm]{\noindent}
6
7 \providecommand{\ebook}{
8 \setcounter{secnumdepth}{0}
9 \setcounter{tocdepth}{2}
10 }

```

File 42 **lwarp-ellipsis.sty**

§ 123 Package **ellipsis**

(Emulates or patches code by PETER J. HESLIN.)

Pkg ellipsis ellipsis is emulated.

```

1 \LWR@ProvidesPackageDrop{ellipsis}

```

```

2
3 \newcommand{\ellipsisgap}{0.1em}

```

File 43 **lwarp-emptypage.sty**

§ 124 Package **emptypage**

Pkg `emptypage` `emptypage` is ignored.

for HTML output: Discard all options for `lwarp-emptypage`:

```

1 \LWR@ProvidesPackageDrop{emptypage}

```

File 44 **lwarp-endnotes.sty**

§ 125 Package **endnotes**

(Emulates or patches code by JOHN LAVAGNINO.)

Pkg `endnotes` Used as-is.

table of contents To place the endnotes in the TOC, use:

```

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

```

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

```

\ForceHTMLPage
\theendnotes

```

for HTML output: `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 45 `lwarp-enumerate.sty`

§ 126 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 46 `lwarp-enumitem.sty`

§ 127 Package **enumitem**

(Emulates or patches code by JAVIER BEZOS.)

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 47 `lwarp-epigraph.sty`§ 128 Package **epigraph***(Emulates or patches code by PETER WILSON.)*

Pkg epigraph epigraph is emulated.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \ifbool{FormatWP}
7 {\begin{BlockClass}[border-top:1px solid gray]{epigraphsource}}
8 {\begin{BlockClass}{epigraphsource}}
9 #2
10 \end{BlockClass}
11 \end{BlockClass}
12 }

13 \newcommand{\epigraph}[2]
14 {
15 \begin{LWR@BlockClassWP}{text-align:right}{\epigraph}
16 \qitem{#1}{#2}
17 \end{LWR@BlockClassWP}
18 }
19
20 \newenvironment*{epigraphs}
21 {\LWR@BlockClassWP{text-align:right}{\epigraph}}
22 {\endLWR@BlockClassWP}

```

Use css to format epigraphs.

The following are null commands for source compatibility:

```

23 \newlength{\epigraphwidth}
24 \setlength{\epigraphwidth}{.5\linewidth}
25 \newenvironment*{flushepinormal}{}{}
26 \newcommand{\textflush}[1]{flushepinormal}
27 \newcommand{\epigraphflush}[1]{flushright}
28 \newcommand{\sourceflush}[1]{flushright}
29 \newcommand*{\epigraphsize}{\small}
30 \newlength{\epigraphrule}

```

```

31 \newlength{\beforeepigraphskip}
32 \newlength{\afterepigraphskip}
33 \newcommand{\epigraphhead}[2][0]{#2}
34 \newcommand{\dropchapter}[1]{}
35 \newcommand*{\undodrop}{}
36 \newcommand{\cleartoevenpage}[1][{}]{}
```

File 48 **lwarp-eso-pic.sty**

§ 129 Package **eso-pic**

(Emulates or patches code by ROLF NIEPRASCHK.)

Pkg `eso-pic` `eso-pic` is emulated.

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 49 **lwarp-everypage.sty**

§ 130 Package **everypage**

(Emulates or patches code by SERGIO CALLEGARI.)

Pkg `everypage` `everypage` is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{everypage}
```

```
2 \newcommand*\AddEverypageHook}[1]{}
3 \newcommand*\AddThispageHook}[1]{}
```

File 50 **lwarp-everyshi.sty**

§ 131 Package **everyshi**

(Emulates or patches code by MARTIN SCHRÖDER.)

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 51 **lwarp-extramarks.sty**

§ 132 Package **extramarks**

(Emulates or patches code by PIET VAN OOSTRUM.)

Pkg extramarks extramarks is emulated.

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 52 **lwarp-fancybox.sty**

§ 133 Package **fancybox**

(Emulates or patches code by TIMOTHY VAN ZANDT.)

Pkg fancybox fancybox is supported with some patches.

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

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

[framing alternatives](#) \fbox works with fancybox. Also see lwarp's \fboxBlock macro and fminipage environment for alternatives to \fbox for framing environments.

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

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

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

```

\newenvironment{FramedVerb}[1] % width
{
\VerbatimEnvironment
\fmnipage{#1}
\beginVerbatim
}{
\endVerbatim
\endfmnipage
}

```

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]{%
4 \ifbool{FormatWP}%
5 {\InlineClass[border:1px solid black]{shadowbox}{#1}}%
6 {\InlineClass{shadowbox}{#1}}%
7 }
8
9 \renewcommand*{\@doublebox}[1]{%
10 \ifbool{FormatWP}%
11 {\InlineClass[border:1px double black]{doublebox}{#1}}%
12 {\InlineClass{doublebox}{#1}}%
13 }
14
15 \renewcommand*{\@ovalbox}[2]{%
16 \ifbool{FormatWP}%
17 {\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
18 {%
19   \ifthenelse{\isequivalentto{#1}{\thinlines}}%
20   {\InlineClass{ovalbox}{#2}}%
21   {\InlineClass{Ovalbox}{#2}}%
22 }%
23 }

```

Convert minipages, parboxes, and lists into linear text using the `LWR@nestspan` environment:

```

24 \let\LWR@origSbox\Sbox
25
26 \def\Sbox{\LWR@origSbox\LWR@nestspan}
27

```

```

28
29 \let\LWR@origendSbox\endSbox
30
31 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

Beqnarray is adapted for MathJax or enclosed inside a lateximage:

```

32 \RenewEnviron{Beqnarray}
33 {\LWR@eqnarrayfactor}
34
35 \csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

\GenericCaption is enclosed in an HTML block:

```

36 \renewcommand{\GenericCaption}[1]{%
37 \LWR@figcaption%
38 #1%
39 \endLWR@figcaption%
40 }

```

Btrivlist is enclosed in an HTML block:

```

41 \RenewDocumentEnvironment{Btrivlist}{m o}
42 {\begin{BlockClass}{Btrivlist}\tabular{#1}}
43 {\endtabular\end{BlockClass}}

```

Btrivlist is also neutralized when used inside a span:

```

44 \AtBeginEnvironment{LWR@nestspan}{%
45 \RenewDocumentEnvironment{Btrivlist}{m o}{\}{}%
46 \RenewDocumentCommand{\LWR@origitem}{d()}{\LWRFB@origitem}%
47 }

```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```

48 \let\LWRFB@origitemizeitem\LWR@itemizeitem
49 \let\LWRFB@origdescitem\LWR@descitem
50 \LetLtxMacro{\LWRFB@origitem}{\LWR@origitem}
51
52 \RenewDocumentCommand{\LWR@itemizeitem}{d()}{\LWRFB@origitemizeitem}
53 \RenewDocumentCommand{\LWR@descitem}{d()}{\LWRFB@origdescitem}

```

The various boxed lists become regular lists:

```

54 \renewenvironment{Bitemize}[1][\]{\begin{itemize}}{\end{itemize}}
55 \renewenvironment{Benumerate}[1][\]{\begin{enumerate}}{\end{enumerate}}
56 \renewenvironment{Bdescription}[1][\]{\begin{description}}{\end{description}}

```

`\boxput` simply prints one then the other argument, side-by-side instead of above and behind:

```
57 \RenewDocumentCommand{\boxput}{s d() m m}{%
58 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
59 }
```

Neutralized commands:

```
60 \RenewDocumentCommand{\fancyput}{s d() m}{%}
61 \RenewDocumentCommand{\thisfancyput}{s d() m}{%}
62
63 \RenewDocumentCommand{\fancypage}{m m}{%}
64 \RenewDocumentCommand{\thisfancypage}{m m}{%}
65
66 \def\LandScape#1{}
67 \def\endLandScape{}
68 \def\@LandScape#1#2#3{}
69 \def\endLandScape{}
```

Low-level patches for `Verbatim`, `VerbatimInput`, `UseVerbatim`:

```
70 \let\LWRFB@UseVerbatim\UseVerbatim
71 \renewcommand*{\UseVerbatim}[1]{%
72 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
73 \LWRFB@UseVerbatim{#1}%
74 \LWR@afterendverbatim%
75 }
76
77 \let\LWRFB@LUseVerbatim\LUseVerbatim
78
79 \renewcommand*{\LUseVerbatim}[1]{%
80 \LWR@atbeginverbatim{LVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
81 \noindent%
82 \LWRFB@LUseVerbatim{#1}%
83 \LWR@afterendverbatim%
84 }
85
86 \def\@BUseVerbatim[#1]#2{%
87 \LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
88 \LWRFB@UseVerbatim{#2}%
89 \LWR@afterendverbatim%
90 }

91 \end{warpHTML}
```

File 53 `lwarp-fancyhdr.sty`§ 134 Package **fancyhdr***(Emulates or patches code by PIET VAN OOSTRUM.)*

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 54 `lwarp-fancyref.sty`§ 135 Package **fancyref**

Pkg fancyref fancyref is emulated.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{fancyref}

2 \newcommand*{\fancyrefhook}[1] {#1}
3 \newcommand*{\fref}[2] [] {\cref{#2}}
4 \newcommand*{\Fref}[2] [] {\Cref{#2}}
5 \newcommand*{\fancyrefchaplabelex}{chap}

```

```
6 \newcommand*\fancyrefenumlabelprefix}{enum}
7 \newcommand*\fancyrefeqlabelprefix}{eq}
8 \newcommand*\fancyreffiglabelprefix}{fig}
9 \newcommand*\fancyreffnlabelprefix}{fn}
10 \newcommand*\fancyrefseclabelprefix}{sec}
11 \newcommand*\fancyreftablabeledprefix}{tab}
12 \newcommand*\fancyrefchange[2]{}
13 \newcommand*\fancyrefargdelim}{:}
14 \newcommand*\fancyrefloosespacing}{-}
15 \newcommand*\fancyreftightspacing}{\,}
16 \newcommand*\fancyrefdefaultspacing}{\fancyrefloosespacing}
17 \providecommand*\chaptername}{Chapter}
18 \providecommand*\figurename}{Figure}
19 \providecommand*\pagename}{Page}
20 \providecommand*\tablename}{Table}
21 \renewcommand*\Frefchapname}{\chaptername}
22 \renewcommand*\Frefenumname}{Item}
23 \renewcommand*\Frefeqname}{Equation}
24 \renewcommand*\Freffigname}{\figurename}
25 \renewcommand*\Freffnname}{Footnote}
26 \renewcommand*\Frefonname}{On}
27 \renewcommand*\Frefpgname}{\pagename}
28 \renewcommand*\Frefsecname}{Section}
29 \renewcommand*\Frefseenname}{See}
30 \renewcommand*\Freftablename}{\tablename}
31 \renewcommand*\Freffigshortname}{Fig.}
32 \renewcommand*\Frefpgshortname}{P.}
33 \renewcommand*\Freftabshortname}{Tab.}
34 \renewcommand*\frefchapname}{\MakeLowercase{\Frefchapname}}
35 \renewcommand*\frefenumname}{\MakeLowercase{\Frefenumname}}
36 \renewcommand*\frefeqname}{\MakeLowercase{\Frefeqname}}
37 \renewcommand*\freffigname}{\MakeLowercase{\Freffigname}}
38 \renewcommand*\freffnname}{\MakeLowercase{\Freffnname}}
39 \renewcommand*\frefonname}{\MakeLowercase{\Frefonname}}
40 \renewcommand*\frefpgname}{\MakeLowercase{\Frefpgname}}
41 \renewcommand*\frefsecname}{\MakeLowercase{\Frefsecname}}
42 \renewcommand*\frefseenname}{\MakeLowercase{\Frefseenname}}
43 \renewcommand*\freftablename}{\MakeLowercase{\Freftablename}}
44 \renewcommand*\freffigshortname}{\MakeLowercase{\Freffigshortname}}
45 \renewcommand*\frefpgshortname}{\MakeLowercase{\Frefpgshortname}}
46 \renewcommand*\freftabshortname}{\MakeLowercase{\Freftabshortname}}
47 \newcommand*\fancyrefaddcaptions[2]{}
48 \newcommand*\fancyrefdefaultformat}{@empty}
49 \newcommand*\frefformat[3]{}
50 \newcommand*\Frefformat[3]{}

```

File 55 `lwarp-fancyvrb.sty`

§ 136 Package **fancyvrb**

(Emulates or patches 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 \newcommand*\LWR@FVfindbordercolor-{}%
52 \FancyVerbRuleColor%
53 \LWR@findcurrenttextcolor%
54 \color{black}%

```

```

55 }
56
57 % border width of \FV@FrameRule
58 \newcommand*{\LWR@FVborderstyle}[1]{%
59 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
60 \LWR@FVfindbordercolor %
61 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ; %
62 }
63
64 \def\FV@Frame@none{%
65 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
66 \let\FV@BeginListFrame\LWR@fvstartnone%
67 \let\FV@LeftListFrame\relax%
68 \let\FV@RightListFrame\relax%
69 \let\FV@endListFrame\LWR@fvendnone}
70
71 \FV@Frame@none% default values
72
73 \def\FV@Frame@single{%
74 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{}}%
75 \let\FV@BeginListFrame\LWR@fvstartsingle%
76 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
77 \let\FV@RightListFrame\FV@RightListFrame@Single%
78 \let\FV@endListFrame\LWR@fvendsingle}
79
80 \def\FV@Frame@lines{%
81 \renewcommand*{\LWR@FVstyle}{%
82   \LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}}%
83 }%
84 \let\FV@BeginListFrame\LWR@fvstartline%
85 \let\FV@LeftListFrame\relax%
86 \let\FV@RightListFrame\relax%
87 \let\FV@endListFrame\LWR@fvendline}
88
89 \def\FV@Frame@topline{%
90 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}%
91 \let\FV@BeginListFrame\LWR@fvstartline%
92 \let\FV@LeftListFrame\relax%
93 \let\FV@RightListFrame\relax%
94 \let\FV@endListFrame\LWR@fvendnone}
95
96 \def\FV@Frame@bottomline{%
97 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}%
98 \let\FV@BeginListFrame\LWR@fvstartnone%
99 \let\FV@LeftListFrame\relax%
100 \let\FV@RightListFrame\relax%
101 \let\FV@endListFrame\LWR@fvendline}
102
103 \def\FV@Frame@leftline{%
104 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}%

```

```

105 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
106 \ifx\FancyVerbFillColor\relax%
107 \let\FV@FrameFillLine\relax%
108 \else%
109 \@tempdima\FV@FrameRule\relax%
110 \multiply\@tempdima-\tw@%
111 \edef\FV@FrameFillLine{%
112 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}%
113 \kern-\number\@tempdima sp}}%
114 \fi%
115 \let\FV@BeginListFrame\LWR@fvstartnone%
116 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
117 \let\FV@RightListFrame\relax%
118 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

119 \def\FV@SingleFrameLine#1{%
120   \hbox to\z@{%
121     \kern\leftmargin
122     \ifnum#1=\z@\relax
123       \let\FV@Label\FV@LabelBegin
124     \else
125       \let\FV@Label\FV@LabelEnd
126     \fi
127     \ifx\FV@Label\relax
128       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
129     \else
130       \ifnum#1=\z@
131         \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
132         \ifx\FV@LabelPositionTopLine\relax
133           \else
134             \LWR@FVfindbordercolor
135             \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
136             \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
137             \LWR@htmltagc{/div}
138             \fi
139           \else
140             \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
141             \ifx\FV@LabelPositionBottomLine\relax
142               \else
143                 \LWR@FVfindbordercolor
144                 \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
145                 \LWR@origtextrm{\FV@LabelEnd}
146                 \LWR@htmltagc{/div}
147                 \fi
148             \fi
149

```

```

150   \fi
151   \hss
152   }
153 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

154 \def\FV@ListProcessLine#1{%
155   \hbox to \hsize{%
156 %     \kern\leftmargin
157     \hbox to \VerbatimHTMLWidth {%
158       \ifcvoid{FV@LeftListNumber}{\kern 2.5em}%
159       \FV@LeftListNumber%
160 %     \FV@LeftListFrame
161     \FancyVerbFormatLine{#1}%
162     \hss%
163 %     \FV@RightListFrame
164     \FV@RightListNumber%
165   }%
166   \hss% required to avoid underfull hboxes
167 }
168 }

```

Env BVerbatim

```

169 \AtBeginEnvironment{BVerbatim}
170 {
171 \LWR@forcenewpage
172 \LWR@atbeginverbatim{bverbatim}
173
174 }
175
176 \AfterEndEnvironment{BVerbatim}
177 {
178 \leavevmode\par\LWR@origvspace{-\baselineskip}
179 \LWR@afterendverbatim
180 }

```

End of the modifications to make at the end of the preamble:

```

181 } % \AfterEndPreamble

182 \end{warpHTML}

```

File 56 **lwarp-figcaps.sty**

§ 137 Package **figcaps**

(Emulates or patches code by PATRICK W. DALY.)

Pkg `figcaps` Emulated.

for HTML output: Discard all options for `lwarp-figcaps`:

```
1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}{}
3 \newcommand*{\figcapsoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}
```

File 57 **lwarp-fix2col.sty**

§ 138 Package **fix2col**

Pkg `fix2col` `fix2col` is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{fix2col}`

File 58 **lwarp-fixme.sty**

§ 139 Package **fixme**

(Emulates or patches code by DIDIER VERNA.)

Pkg `fixme` `fixme` is patched for use by `lwarp`.

 **external layouts** External layouts (`\fxloadlayouts`) are not supported.

User control is provided for setting the HTML styling of the “faces”. The defaults are as follows, and may be changed in the preamble after `fixme` is loaded:

```

\def\FXFaceInlineHTMLStyle{font-weight:bold}
\def\FXFaceEnvHTMLStyle{font-weight:bold}
\def\FXFaceSignatureHTMLStyle{font-style:italic}
\def\FXFaceTargetHTMLStyle{font-style:italic}

```

for HTML output: 1 \LWR@ProvidesPackagePass{fixme}

Restore lwarp's version of \@wrindex, ignoring the fixme package's target option:

```
2 \let\@wrindex\LWR@wrindex
```

Float-related macros required by lwarp:

```

3 \newcommand{\ext@fixme}{lox}
4
5 \renewcommand{\l@fixme}[2]{\hypertocfloat{1}{fixme}{lox}{#1}{#2}}

```

Other modifications:

```

6 \def\FXFaceInlineHTMLStyle{font-weight:bold}
7
8 \renewcommand*\FXLayoutInline[3]{ %
9 \InlineClass[\FXFaceInlineHTMLStyle]{fixmeinline}%
10 {\@fxttextstd{#1}{#2}{#3}}%
11 }
12
13 \def\FXFaceEnvHTMLStyle{font-weight:bold}
14
15 \renewcommand*\FXEnvLayoutPlainBegin[2]{%
16 \BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
17 \ignorespaces#2 \fxnotename{#1}: \ignorespaces}
18
19 \renewcommand*\FXEnvLayoutPlainEnd[2]{\endBlockClass}
20
21 \renewcommand*\FXEnvLayoutSignatureBegin[2]{%
22 \BlockClass[\FXFaceEnvHTMLStyle]{fixmebold}
23 \fxnotename{#1}: \ignorespaces}
24
25 \renewcommand*\FXEnvLayoutSignatureEnd[2]{\@fxsignature{#2}\endBlockClass}
26
27 \def\FXFaceSignatureHTMLStyle{font-style:italic}
28
29 \DeclareRobustCommand*\@fxsignature[1]{%
30 \ifthenelse{\equal{#1}{}}%
31 {}%
32 { -- {\InlineClass[\FXFaceSignatureHTMLStyle]{fixmesignature}{#1}}}%
33 }
34
35

```

```

36 \def\FXFaceTargetHTMLStyle{font-style:italic}
37
38 \renewcommand\FXTargetLayoutPlain[2]{%
39   \InlineClass[\FXFaceTargetHTMLStyle]{fixmetarget}{#2}%
40 }

```

File 59 **lwarp-fixmetodonotes.sty**

§ 140 Package **fixmetodonotes**

(Emulates or patches code by GIOELE BARABUCCI.)

Pkg `fixmetodonotes` `fixmetodonotes` is patched for use by `lwarp`.

for HTML output:

```

1 \LWR@ProvidesPackagePass{fixmetodonotes}

2 \renewcommand{\NOTES@addtolist}[2]{%
3   \refstepcounter{NOTES@note}%
4 %   \phantomsection% REMOVED
5   \addcontentsline{notes}{NOTES@note}{%
6     \protect\numberline{\theNOTES@note}{{#1}: {#2}}%
7   }%
8 }
9
10 \renewcommand{\NOTES@marker}[2]{\fbox{%
11   \textcolor{#2}{% WAS \color
12     \textbf{#1}}%
13   }}
14
15 \renewcommand{\NOTES@colorline}[2]{%
16   \bgroup%
17   \ULon{\LWR@backgroundcolor{#1}{#2}}%
18 }

```

File 60 **lwarp-float.sty**

§ 141 Package **float and \newfloat**

(Emulates or patches code by ANSELM LINGNAU.)

Pkg `float` `float` is emulated.

for HTML output: `1 \LWR@ProvidesPackageDrop{float}[2016/03/04]`

See section [64.2](#) for the `\listof` command.

`\newfloat` $\langle 1: type \rangle \langle 2: placement \rangle \langle 3: ext \rangle [\langle 4: within \rangle]$

Emulates the `\newfloat` command from the `float` package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}
4 {\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}
5 {\DeclareFloatingEnvironment[fileext=#3]{#1}}
```

`newfloat` package automatically creates the `\listof` command for new floats, but `float` does not, so remove `\listof` here in case it is manually created later.

```
6 \cslet{listof#1s}\relax
7 \cslet{listof#1es}\relax
8 }
```

`\floatname` $\langle type \rangle \langle name \rangle$

Sets the text name of the float, such as “Figure”.

```
9 \NewDocumentCommand{\floatname}{m +m}{%
10 \SetupFloatingEnvironment{#1}{name=#2}%
11 }
```

`\floatplacement` $\langle type \rangle \langle placement \rangle$

Float placement is ignored.

```
12 \newcommand*{\floatplacement}[2]{%
13 \SetupFloatingEnvironment{#1}{placement=#2}%
14 }
```

`\floatstyle` $\langle style \rangle$

Float styles are ignored.

```
15 \newcommand{\floatstyle}[1]{%
16 }
```

`\restylefloat` * $\langle style \rangle$

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%
18 }
```

File 61 **lwarp-floatflt.sty**

§ 142 Package **floatflt**

(Emulates or patches code by MATS DAHLGREN.)

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Env [$\langle \rangle$] offset $\langle \langle type \rangle \rangle$ $\langle \langle width \rangle \rangle$ Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m m}
3 {%
4 \setlength{\LWR@templengthone}{#3}%
5 \uselengthunit{PT}%
6 \LWR@BlockClassWP{%
7   float:right; %
8   width:\rndprintlength{\LWR@templengthone}; %
9   margin:10pt%
10 }{%
11   width:\rndprintlength{\LWR@templengthone}%
12 }%
13 {marginblock}%
14 \captionsetup{type=#2}%
15 }
16 {%
17 \endLWR@BlockClassWP%
18 }
```

Env floatingfigure [$\langle placement \rangle$] $\langle \langle width \rangle \rangle$

```
19 \DeclareDocumentEnvironment{floatingfigure}{o m}
20 {\begin{KFLTfloatflt@marginfloat}{figure}{#2}}
21 {\end{KFLTfloatflt@marginfloat}}
```

Env floatingtable [$\langle placement \rangle$]

```
22 \DeclareDocumentEnvironment{floatingtable}{o}
23 {\begin{KFLTfloatflt@marginfloat}{table}{1.5in}}
24 {\end{KFLTfloatflt@marginfloat}}
```

File 62 **lwarp-floatpag.sty**§ 143 Package **floatpag**

(Emulates or patches code by VYTAS STATULEVIČIUS AND SIGITAS TOLUŠIS.)

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 63 **lwarp-floatrow.sty**§ 144 Package **floatrow**

(Emulates or patches code by OLGA LAPKO.)

Pkg floatrow floatrow is emulated.

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}{}

```

`\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` $[\langle width \rangle]$ $[\langle height \rangle]$ $[\langle vposn \rangle]$ $\langle caption \text{ commands} \rangle$ $\langle contents \rangle$

```
67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []
```

`\ttabbox` $[\langle width \rangle]$ $[\langle height \rangle]$ $[\langle vposn \rangle]$ $\langle caption \text{ commands} \rangle$ $\langle contents \rangle$

```
68 \newfloatcommand{ttabbox}{table}[\capttop] [\FBwidth]
```

`\fcapside` $[\langle width \rangle]$ $[\langle height \rangle]$ $[\langle vposn \rangle]$ $\langle caption \text{ commands} \rangle$ $\langle contents \rangle$

```
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 64 **lwarp-flushend.sty**

§ 145 Package **flushend**

(Emulates or patches code by SIGITAS TOLUŠIS.)

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 65 **lwarp-fncychap.sty**

§ 146 Package **fncychap**

(Emulates or patches code by ULF A. LINDGREN.)

Pkg fncychap fncychap is emulated.

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 66 **lwarp-fnpos.sty**

§ 147 Package **fnpos**

(Emulates or patches code by HIROSHI NAKASHIMA.)

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 67 **lwarp-fontenc.sty**

§ 148 Package **fontenc**

Pkg fontenc If using pdf_{La}T_EX, lwarp used to require fontspec be loaded before lwarp, but now lwarp itself loads \fontspec with T1 encoding, which lwarp requires. fontspec is now allowed to be loaded with another encoding after lwarp.

lwarp-fontenc is no longer necessary, but is still provided to overwrite older versions.

for HTML output:

```

1 \LWR@ProvidesPackagePass{fontenc}

```

File 68 **lwarp-fontspec.sty**

§ 149 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 69 **lwarp-footmisc.sty**

§ 150 Package **footmisc**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg footmisc footmisc is emulated.

```
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 70 `lwarp-footnote.sty`

§ 151 Package **footnote**

(Emulates or patches code by MARK WOODING.)

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   \begingroup%
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 71 **lwarp-footnotehyper.sty**

§ 152 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 72 **lwarp-framed.sty**

§ 153 Package **framed**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg framed framed is supported and patched by lwarp.

for HTML output: Accept all options for lwarp-framed:

```

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}{-}{Opt}{Opt}{Opt}{Opt}{\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 73 **lwarp-ftnright.sty**

§ 154 Package **ftnright**

Pkg ftnright ftnright is ignored.

for HTML output: Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

File 74 **lwarp-fullpage.sty**

§ 155 Package **fullpage**

Pkg fullpage fullpage is ignored.

for HTML output: Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

File 75 **lwarp-fullwidth.sty**

§ 156 Package **fullwidth**

(Emulates or patches code by MARCO DANIEL.)

Pkg fullwidth fullwidth is emulated.

A minipage is used, of no HTML width.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fullwidth}

2 \newenvironment*{fullwidth}[1][]{%
3 \minipagefullwidth%
4 \minipage{\linewidth}%
5 }
6 {%
7 \endminipage%
8 }
```

File 76 **lwarp-geometry.sty**

§ 157 Package **geometry**

(Emulates or patches code by HIDEO UMEKI.)

Pkg geometry geometry is preloaded by lwarp, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for lwarp-geometry:

```

1 \LWR@ProvidesPackageDrop{geometry}

2 \renewcommand*\geometry[1]{}
3 \renewcommand*\newgeometry[1]{}
4 \renewcommand*\restoregeometry{}
5 \renewcommand*\savegeometry[1]{}
6 \renewcommand*\loadgeometry[1]{}

```

File 77 `lwarp-glossaries.sty`§ 158 Package **glossaries**

(Emulates or patches code by NICOLA L.C. TALBOT.)

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`

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 78 `lwarp-graphics.sty`

§ 159 Package **graphics**

(Emulates or patches code by D. P. CARLISLE.)

Pkg graphics graphics is emulated.

for HTML output: `1 \LWR@ProvidesPackagePass{graphics}`

§ 159.1 **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 }
```

§ 159.2 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@igyyscale{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@igyyscale}{#1}}
```

Numerous ignored keys:

```
73 \define@key{igraph}{bb}{}
74 \define@key{igraph}{bbllx}{}
75 \define@key{igraph}{bbly}{}
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}{}

```

§ 159.3 Printing HTML styles

```
\LWR@rotstyle {<prefix>} {<degrees>}
```

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```
91 \newcommand*\LWR@rotstyle}[2]{%
92 #1transform:rotate(-#2deg);
93 }
```

`\LWR@scalestyle` $\langle prefix \rangle$ $\langle xscale \rangle$ $\langle yscale \rangle$

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 }
```

§ 159.4 `\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}
```

Used to determine the actual image size if needed:

```
100 \newsavebox{\LWR@imagesizebox}

101 \let\LWR@origGin@setfile\Gin@setfile
```

Define the new class key for the print-mode version of `\includegraphics`, which is enabled inside a `lateximage`.

```
102 \AtBeginDocument{
103 \define@key{Gin}{class}{}
104 }
```

`\LWR@includegraphicsb` * $[\langle 2: options \rangle]$ $[\langle 3: options \rangle]$ $\langle 4: filename \rangle$

graphics syntax is `\includegraphics * $[\langle llx, lly \rangle]$ $[\langle urx, ury \rangle]$ $\langle file \rangle$`

graphicx syntax is `\includegraphics $[\langle key values \rangle]$ $\langle file \rangle$`

If #3 is empty, only one optional argument was given, thus graphicx syntax.

```
105 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
106 {%
107 \LWR@traceinfo{LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
108 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
109 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
110 {%
111   \ifbool{LWR@infloatrow}%
112   {}
113   {% not in a minipage or a floatrow:
114     \setlength{\linewidth}{6in}%
115     \setlength{\textwidth}{6in}%
116     \setlength{\textheight}{9in}%
117   }%
118 }{}%

119 \begingroup%
120 \renewcommand*{\Gin@setfile}[3]{%
121 \LWR@traceinfo{Gin@setfile ##3}%
122 \xdef\LWR@parsedfilename{##3}%
123 }%
124 \Gininclude@graphics{\detokenize\expandafter{##4}}%
125 \endgroup%
126 \filename@parse{\LWR@parsedfilename}%
```

For correct em sizing during the width and height conversions:

```
127 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```
128 \setlength{\LWR@igwidth}{0pt}%
129 \setlength{\LWR@igheight}{0pt}%
130 \renewcommand*{\LWR@igwidthstyle}{}%
131 \renewcommand*{\LWR@igheightstyle}{}%
132 \renewcommand*{\LWR@igorigin}{}%
133 \renewcommand*{\LWR@igangle}{}%
134 \renewcommand*{\LWR@igxscale}{1}%
135 \renewcommand*{\LWR@igyyscale}{1}%
136 \renewcommand*{\LWR@igclass}{inlineimage}%
```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```
137 \IfValueF{#3}{%
138 \IfValueTF{#2}%
139 {\setkeys{igraph}{#2}}%
```

```
140 {\setkeys{igraph}{}}%
141 }%
```

If formatting for a word processor, find and set the actual image size, without rotation, using PDF instead of SVG to find the original bounding box:

```
142 \ifbool{FormatWP}{%
143   \begingroup%
144   \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
145   \define@key{Gin}{angle}{}%
146   \IfBooleanTF{#1}%
147   {% starred
148     \IfValueTF{#3}%
149     {%
150       \global\setbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] [#3] {#4}}%
151     }%
152     {%
153       \IfValueTF{#2}%
154       {%
155         \global\setbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] {#4}}%
156       }%
157       \global\setbox{\LWR@imagesizebox}{\LWR@originincludegraphics* {#4}}%
158     }%
159   }%
160 }% starred
161 }% not starred
162   \IfValueTF{#3}%
163   {%
164     \global\setbox{\LWR@imagesizebox}{\LWR@originincludegraphics [#2] [#3] {#4}}%
165   }%
166   {%
167     \IfValueTF{#2}%
168     {%
169       \global\setbox{\LWR@imagesizebox}{\LWR@originincludegraphics [#2] {#4}}%
170     }%
171     \global\setbox{\LWR@imagesizebox}{\LWR@originincludegraphics {#4}}%
172   }%
173 }%
174 }% not starred
175 \endgroup%
176 \uselengthunit{PT}%
177 \settoheight{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
178 \global\renewcommand*{\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
179 \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
180 \global\renewcommand*{\LWR@igheightstyle}{height:\rndprintlength{\LWR@igheight}}%
181 }{}%
```

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.

```

182 \LWR@traceinfo{\LWR@includegraphicsb: about to create href}%
183 \href{\LWR@parsedfilename}%
184 {% start of href
185 \LWR@traceinfo{\LWR@includegraphicsb: about to LWR@htmltag}%
186 \LWR@htmltag{% start of image tags
187 img src="\LWR@parsedfilename" \LWR@orignewline%
188 \LWR@origtilde{} alt="[filename@base]" \LWR@orignewline%

```

Only include a style tag if a width, height, angle, or scale was given:

```

189 \ifthenelse{
190   \NOT\equal{\LWR@igwidthstyle}{}} \OR
191   \NOT\equal{\LWR@igheightstyle}{}} \OR
192   \NOT\equal{\LWR@igorigin}{}} \OR
193   \NOT\equal{\LWR@igangle}{}} \OR
194   \NOT\equal{\LWR@igxscale}{1}} \OR
195   \NOT\equal{\LWR@igyyscale}{1}}
196 }%
197 {\LWR@origtilde{} style="%
198 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
199 {\LWR@igwidthstyle;}}}%
200 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
201 {\LWR@igheightstyle;}}}%
202 \ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
203 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}}%
204 \ifthenelse{\NOT\equal{\LWR@igangle}{}}%
205 {%
206 \LWR@rotstyle{-ms-}{\LWR@igangle}%
207 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
208 \LWR@rotstyle{}{\LWR@igangle%
209 }}}%
210 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}}\OR%
211 \NOT\equal{\LWR@igyyscale}{1}}%
212 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyyscale}%
213 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyyscale}%
214 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyyscale}}}%
215 %
216 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
217 {opacity:\LWR@opacity;}}%
218 {}%
219 %
220 " \LWR@orignewline}{}}%

```

Set the class:

```

221 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%

```

```

222 }% end of image tags
223 }% end of href
224 \endgroup

```

Return to small-sized output:

```

225 \LWR@origscriptsize
226 \LWR@traceinfo{LWR@includegraphicsb done}%
227 }

```

`\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.

```

228 \AtBeginDocument{
229
230 \LWR@traceinfo{Patching includegraphics.}
231
232 \LetLtxMacro\LWR@originincludegraphics\includegraphics
233
234 \renewcommand*\includegraphics}
235 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

236 \LWR@traceinfo{includegraphics}
237 \LWR@ensuredoingapar%
238 \begingroup%
239 \catcode'\_ =12
240 \LWR@includegraphicsb%
241 }% includegraphics
242 }% AtBeginDocument

```

§ 159.5 Boxes

`\LWR@rotboxorigin` Holds the origin key letters.

```

243 \newcommand*\LWR@rotboxorigin}{

```

`\LWR@originname` {*letter*}

Given one \TeX origin key value, translate into an HTML origin word:

```

244 \newcommand*\LWR@originname}[1]{%
245 \ifthenelse{\equal{#1}{t}}{top}{}%
246 \ifthenelse{\equal{#1}{b}}{bottom}{}%
247 \ifthenelse{\equal{#1}{c}}{center}{}%
248 \ifthenelse{\equal{#1}{l}}{left}{}%
249 \ifthenelse{\equal{#1}{r}}{right}{}%
250 }

```

`\LWR@originnames` $\{ \langle letters \rangle \}$

Given one- or two-letter L^AT_EX origin key values, translate into HTML origin words:

```

251 \newcommand*\LWR@originnames}[1]{%
252 \StrChar{#1}{1}[\LWR@strresult]%
253 \LWR@originname{\LWR@strresult}
254 \StrChar{#1}{2}[\LWR@strresult]%
255 \LWR@originname{\LWR@strresult}
256 }

```

Handle the origin key for `\rotatebox`:

```

257 \define@key{krotbox}{origin}{%
258 \renewcommand*\LWR@rotboxorigin{#1}%
259 }

```

These keys are ignored:

```

260 \define@key{krotbox}{x}{}
261 \define@key{krotbox}{y}{}
262 \define@key{krotbox}{units}{}

```

`\rotatebox` $[\langle keyval list \rangle] \{ \langle angle \rangle \} \{ \langle text \rangle \}$

```

263 \LetLtxMacro\LWR@origrotatebox\rotatebox
264
265 \AtBeginDocument{
266 \RenewDocumentCommand{\rotatebox}{O{} m +m}{%

```

Reset the origin to “none-given”:

```

267 \renewcommand*\LWR@rotboxorigin{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```

268 \setkeys{krotbox}{#1}%

```

Select inline-block so that HTML will transform this span:

```
269 \LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
270 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{}}%
271 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%
```

Print the rotation information:

```
272 \LWR@rotstyle{-ms-}{#2} %
273 \LWR@rotstyle{-webkit-}{#2} %
274 \LWR@rotstyle{}{#2} %
275 "{}\LWR@orignewline%
```

Print the text to be rotated:

```
276 \begin{LWR@nestspan}%
277 #3%
```

Close the span:

```
278 \LWR@htmltagc{/span}%
279 \end{LWR@nestspan}%
280 }
281 }% AtBeginDocument
```

`\scalebox` $\langle h\text{-scale} \rangle$ $[\langle v\text{-scale} \rangle]$ $\langle text \rangle$

```
282 \LetLtxMacro\LWR@origscalebox\scalebox
283
284 \AtBeginDocument{
285 \RenewDocumentCommand{\scalebox}{m o m}{%
```

Select inline-block so that HTML will transform this span:

```
286 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
287 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
288 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
289 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
290 "{}}%
```

Print the text to be scaled:

```
291 \begin{LWR@nestspan}%
292 #3%
```

Close the span:

```
293 \LWR@htmltagc{/span}%
294 \end{LWR@nestspan}%
295 }
296 }% AtBeginDocument
```

`\reflectbox` $\{ \langle text \rangle \}$

```
297 \let\LWR@origreflectbox\reflectbox
298
299 \AtBeginDocument{
300 \renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
301 }
```

`\resizebox` $\{ \langle h-length \rangle \} \{ \langle v-length \rangle \} \{ \langle text \rangle \}$

Simply prints its text argument.

```
302 \LetLtxMacro\LWR@origresizebox\resizebox
303
304 \AtBeginDocument{
305 \renewcommand{\resizebox}[3]{#3}
306 }
```

File 79 **lwarp-graphics.sty**

§ 160 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 80 **lwarp-grffile.sty**

§ 161 Package **grffile**

Pkg `grffile` `grffile` is supported as-is. File types known to the browser are displayed, and unknown

 **matching PDF and SVG** file types are given a link. Each PDF image for print mode should be accompanied by an SVG, PNG, or JPG version for HTML.

lwarp-grffile now exists as a placeholder since grffile used to be emulated by lwarp, and thus older versions of lwarp-grffile may exist and should be overwritten by this newer version.

for HTML output: 1 \LWR@ProvidesPackagePass{grffile}

File 81 **lwarp-hang.sty**

§ 162 Package **hang**

Pkg hang hang is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{hang}

2 \newlength{\hangingindent}
3 \setlength{\hangingindent}{1em}
4 \newlength{\hangingleftmargin}
5 \setlength{\hangingleftmargin}{0em}
6
7 \newcommand*\LWR@findhangingleftmargin{%
8 \setlength{\LWR@templengthone}{\hangingleftmargin}%
9 \addtolength{\LWR@templengthone}{\hangingindent}%
10 }
11
12 \newenvironment{hangingpar}
13 {
14   \LWR@findhangingleftmargin%
15   \uselengthunit{PT}%
16   \BlockClass [%
17     margin-left:\rndprintlength{\LWR@templengthone}; %
18     text-indent:-\rndprintlength{\hangingindent}%
19   ]%
20   {hangingpar}%
21 }
22 {\endBlockClass}
23
24 \newenvironment{hanginglist}
25 {%
26   \renewcommand*\LWR@printcloselist{\LWR@printcloseitemize}%
27   \renewcommand*\LWR@printopenlist{%
28     \LWR@findhangingleftmargin%
29     \uselengthunit{PT}%
30     ul style="list-style-type:none; %
31     margin-left:\rndprintlength{\LWR@templengthone}; %

```

```

32     text-indent:-\rndprintlength{\hangingindent}"%
33   }%
34   \let\item\LWR@itemizeitem%
35   \list{}{}%
36 }
37 {\endlist}
38
39 \newenvironment{compacthang}
40 {\hanginglist}
41 {\endhanginglist}
42
43 \newlength{\labeledleftmargin}
44 \setlength{\labeledleftmargin}{0em}
45
46 \newenvironment{labeledpar}[2]
47 {%
48   \BlockClass[%
49     \LWR@findhangingleftmargin%
50     \uselengthunit{PT}]%
51     margin-left:\rndprintlength{\LWR@templengthone}; %
52     text-indent:-\rndprintlength{\hangingindent}%
53   ][labeledpar]#2%
54 }
55 {\endBlockClass}
56
57 \newenvironment{labeledlist}[1]
58 {\hanginglist}
59 {\endhanginglist}
60
61 \newenvironment{compactlabel}[1]
62 {\hanginglist}
63 {\endhanginglist}

```

File 82 **lwarp-hyperref.sty**

§ 163 Package **hyperref**

(Emulates or patches code by SEBASTIAN RAHTZ, HEIKO OBERDIEK.)

Pkg hyperref hyperref is emulated.

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` $\{\langle url \rangle\} \{\langle alt text \rangle\}$

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 \DeclareRobustCommand*\hyperimage}{%
17 \LWR@ensuredoingapar%
18 \begingroup\catcode'\_ =12
19 \LWR@hyperimageb%
20 }
```

`\hyperdef` $\{\langle 1: category \rangle\} \{\langle 2: name \rangle\} \{\langle 3: text \rangle\}$

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` $\{\langle 1: URL \rangle\} \{\langle 2: category \rangle\} \{\langle 3: name \rangle\} \{\langle 4: text \rangle\}$

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}{O{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 \DeclareRobustCommand*\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` $\{\langle macroname \rangle\} \{\langle T\textsubscript{E}Xstring \rangle\}$

```
56 \newcommand{\pdfstringdef}[2]{}
```

```

\pdfbookmark [level] {text} {name}
57 \newcommand{\pdfbookmark}[3] [] {}

\currentpdfbookmark {text} {name}
58 \newcommand{\currentpdfbookmark}[2] {}

\subpdfbookmark {text} {name}
59 \newcommand{\subpdfbookmark}[2] {}

\belowpdfbookmark {text} {name}
60 \newcommand{\belowpdfbookmark}[2] {}

\texorpdfstring {TEXstring} {PDFstring}
61 \newcommand{\texorpdfstring}[2] {#2}

\hypercalcbp {dimen} From hyperref.
62 \def\hypercalcbp#1{%
63 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
64 }%

\Acrobatmenu {menuoption} {text}
65 \newcommand{\Acrobatmenu}[2] {}

\TextField [parameters] {label}
66 \DeclareRobustCommand{\TextField}[2] [] {}

\CheckBox [parameters] {label}
67 \DeclareRobustCommand{\CheckBox}[2] [] {}

\ChoiceMenu [parameters] {label} {choices}
68 \DeclareRobustCommand{\ChoiceMenu}[3] [] {}

\PushButton [parameters] {label}
69 \DeclareRobustCommand{\PushButton}[2] [] {}

```

<code>\Submit</code>	<code>[\parameters] {\label}</code>
	70 <code>\DeclareRobustCommand{\Submit}[2] [] {}</code>
<code>\Reset</code>	<code>[\parameters] {\label}</code>
	71 <code>\DeclareRobustCommand{\Reset}[2] [] {}</code>
<code>\Gauge</code>	<code>[\parameters] {\label}</code>
	72 <code>\DeclareRobustCommand{\Gauge}[2] [] {}</code>
<code>\LayoutTextField</code>	<code>{\label} {\field}</code>
	73 <code>\newcommand*{\LayoutTextField}[2] {}</code>
<code>\LayoutChoiceField</code>	<code>{\label} {\field}</code>
	74 <code>\newcommand*{\LayoutChoiceField}[2] {}</code>
<code>\LayoutCheckField</code>	<code>{\label} {\field}</code>
	75 <code>\newcommand*{\LayoutCheckField}[2] {}</code>
<code>\MakeRadioField</code>	<code>{\width} {\height}</code>
	76 <code>\newcommand*{\MakeRadioField}[2] {}</code>
<code>\MakeCheckField</code>	<code>{\width} {\height}</code>
	77 <code>\newcommand*{\MakeCheckField}[2] {}</code>
<code>\MakeTextField</code>	<code>{\width} {\height}</code>
	78 <code>\newcommand*{\MakeTextField}[2] {}</code>
<code>\MakeChoiceField</code>	<code>{\width} {\height}</code>
	79 <code>\newcommand*{\MakeChoiceField}[2] {}</code>
<code>\MakeFieldButton</code>	<code>{\text}</code>
	80 <code>\newcommand{\MakeFieldButton}[1] {}</code>

File 83 `lwarp-hyperxmp.sty`

§ 164 Package **hyperxmp**

Pkg hyperxmp Emulated.

for HTML output: Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 84 `lwarp-idxlayout.sty`

§ 165 Package **idxlayout**

(Emulates or patches code by THOMAS TITZ.)

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 85 `lwarp-ifoddpagel.sty`

§ 166 Package **ifoddpagel**

(Emulates or patches code by MARTIN SCHARRER.)

Pkg `ifoddpagel` `ifoddpagel` is emulated.

for HTML output: Discard all options for `lwarp-ifoddpagel`:

```

1 \LWR@ProvidesPackageDrop{ifoddpagel}

2 \newif\ifoddpagel
3
4 \newif\ifoddpageloneside
5
6 \DeclareRobustCommand{\checkoddpagel}{\oddpageltrue\oddpagelonesidetrue}
7
8 \def\oddpagel@pagel{1}
9
10 \def\@ifoddpagel{%
11     \expandafter\@firstoftwo
12 }
13
14 \def\@ifoddpageloneside{%
15     \expandafter\@firstoftwo
16 }
```

File 86 `lwarp-indentfirst.sty`

§ 167 Package **indentfirst**

Pkg `indentfirst` `indentfirst` is ignored.

Discard all options for `lwarp-indentfirst`:

for HTML output: `1 \LWR@ProvidesPackageDrop{indentfirst}`

File 87 `lwarp-inputenc.sty`

§ 168 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 88 `lwarp-keyfloat.sty`

§ 169 Package **keyfloat**

(Emulates or patches code by BRIAN DUNN.)

Pkg `keyfloat` `keyfloat` is supported with minor adjustments.

 **keywrap** If placing a `\keyfig[H]` inside a `keywrap`, use an absolute width for `\keyfig`, instead of `lw`-proportional widths. (The `[H]` option forces the use of a minipage, which internally adjusts for a virtual 6-inch wide minipage, which then corrupts the `lw` option.)

for HTML output:

```
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{KFLT@boxinner: done}%
13 }
```

```

14 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
15 {%
16 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{-}{marginblock}%
17 \captionsetup{type=#2}%
18 }
19 {%
20 \endLWR@BlockClassWP%
21 }

22 \DeclareDocumentEnvironment{marginfigure}{o}
23 {\begin{KFLT@marginfloat}{figure}}
24 {\end{KFLT@marginfloat}}
25
26 \DeclareDocumentEnvironment{margintable}{o}
27 {\begin{KFLT@marginfloat}{table}}
28 {\end{KFLT@marginfloat}}

29 \DeclareDocumentEnvironment{keywrap}{m +m}
30 {%
31 \LWR@ensuredoingapar%
32 \setlength{\LWR@templengthone}{#1}%
33 \uselengthunit{PT}%
34 \begin{LWR@BlockClassWP}{%
35   float:right; width:\rndprintlength{\LWR@templengthone}; %
36   margin:10pt%
37 }%
38 {%
39   width:\rndprintlength{\LWR@templengthone}%
40 }%
41 {marginblock}%
42 \setlength{\linewidth}{.95\LWR@templengthone}%
43 #2%
44 \end{LWR@BlockClassWP}%
45 }
46 {%
47 }

48 }% AtBeginDocument

```

File 89 **lwarp-layout.sty**

§ 170 Package **layout**

(Emulates or patches code by KENT MCPHERSON, JOHANNES BRAAMS, HIDEO UMEKI.)

Pkg layout layout is emulated.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

File 90 **lwarp-letterspace.sty**

§ 171 Package **letterspace**

(Emulates or patches code by R SCHLICHT.)

Pkg letterspace letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}

2 \newcommand*\lsstyle{}
3 \newcommand\textls[2][1]{}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

File 91 **lwarp-lettrine.sty**

§ 172 Package **lettrine**

(Emulates or patches 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 92 **lwarp-lips.sty**

§ 173 Package **lips**

(Emulates or patches code by MATT SWIFT.)

Pkg lips lips is emulated.

```

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 93 **lwarp-listings.sty**

§ 174 Package **listings**

(Emulates or patches code by CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN.)

Pkg listings listings is supported with some limitations. Text formatting is not yet supported.

for HTML output: 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@origlsthk@EveryPar \lsthk@EveryPar
7
8 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{lstlisting}{1ol}{#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@origlsthk@EveryPar%
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 reenable line numbers:

```
27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterrendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%
```

Inline, so create the closing :

```
32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%
```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

`\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 #1%
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\lst@name\lst@name
50 \lst@ifno101\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 \@tempskipa}%
74 %             \def\lst@vskip{\nobreak\@vskip\@tempskipa\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 94 `lwarp-longtable.sty`

§ 175 Package **longtable**

(Emulates or patches code by DAVID CARLISLE.)

Pkg `longtable` `longtable` is emulated.

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` * [*horizontalment*] [*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 95 **lwarp-lscape.sty**

§ 176 Package **lscape**

(Emulates or patches code by D. P. CARLISLE.)

Pkg `lscape` `lscape` is emulated.

for HTML output: Discard all options for `lwarp-lscape`.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

File 96 **lwarp-ltcaption.sty**

§ 177 Package **ltcaption**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg `ltcaption` `ltcaption` is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{ltcaption}
```

`\LTcapttype` 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 97 `lwarp-ltxtable.sty`

§ 178 Package **ltxtable**

Pkg `ltxtable` `ltxtable` is emulated.

for HTML output: `1 \LWR@ProvidesPackageDrop{ltxtable}`

```
\LTXtable {<width>} {<file>}
2 \newcommand*{\LTXtable}[2]{%
3 \input{#2}%
4 }
```

File 98 `lwarp-luatodonotes.sty`

§ 179 Package **luatodonotes**

(Emulates or patches code by FABIAN LIPP.)

Pkg `luatodonotes` `luatodonotes` is emulated.

The documentation for `todonotes` and `luatodonotes` have an example with a `todo` inside a caption. If this example does not work it will be necessary to move the `todo` outside of the caption.

for HTML output: `1 \LWR@ProvidesPackagePass{luatodonotes}`

Nullify options:

```
2 \@todonotes@additionalMarginEnabledfalse
```

```

3 \if@todonotes@disabled
4 \else
5
6 \newcommand{\ext@todo}{tdo}
7
8 \renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}
9
10
11 \renewcommand{\@todonotes@drawMarginNoteWithLine}{%
12 \fcolorbox
13   {\@todonotes@currentbordercolor}
14   {\@todonotes@currentbackgroundcolor}
15   {\arabic{\@todonotes@numberoftodonotes}}
16 \marginpar{\@todonotes@drawMarginNote}
17 }
18
19 \renewcommand{\@todonotes@drawInlineNote}{%
20 \fcolorboxBlock%
21   {\@todonotes@currentbordercolor}%
22   {\@todonotes@currentbackgroundcolor}%
23   {%
24     \if@todonotes@authorgiven%
25     {\@todonotes@author:\,}%
26     \fi%
27     \@todonotes@text%
28   }%
29 }
30
31 \newcommand{\@todonotes@drawMarginNote}{%
32   \if@todonotes@authorgiven%
33     \@todonotes@author\par%
34   \fi%
35   \arabic{\@todonotes@numberoftodonotes}: %
36   \fcolorbox%
37     {\@todonotes@currentbordercolor}%
38     {\@todonotes@currentbackgroundcolor}%
39     {%
40       \@todonotes@sizecommand%
41       \@todonotes@text %
42     }%
43 }%
44
45 \renewcommand{\missingfigure}[2] [] {%
46 \setkeys{todonotes}{#1}%
47 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
48 \fcolorboxBlock%
49   {\@todonotes@currentbordercolor}%
50   {\@todonotes@currentfigcolor}%
51   {%
52     \setlength{\fboxrule}{4pt}%

```

```

53     \fcolorbox{red}{white}{Missing figure} \quad #2%
54   }
55 }
56
57 \LetLtxMacro\LWRTODONOTES@orig@todocommon\@todocommon
58
59 \RenewDocumentCommand{\@todocommon}{m m}{%
60 \begingroup%
61 \renewcommand*\phantomsection}{}%
62 \LWRTODONOTES@orig@todocommon{#1}{#2}%
63 \endgroup%
64 }
65
66 \renewcommand{\@todoarea}[3] [] {%
67   \@todonotes@areaselectedtrue%
68   \@todocommon{#1}{#2}%
69   \todonotes@textmark@highlight{#3}%
70   \zref@label{\@todonotes@arabic{\@todonotes@numberoftodonotes}@end}%
71 }%
72
73
74 \DeclareDocumentCommand{\todonotes@textmark@highlight}{m}{%
75 \InlineClass[background:\#B3FFB3]{highlight}{#1}%
76 }
77
78 \fi% \if@todonotes@disabled

```

File 99 **lwarp-marginfit.sty**

§ 180 Package **marginfit**

Pkg marginfit marginfit is ignored.

for HTML output: Discard all options for lwarp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}
```

File 100 **lwarp-marginfix.sty**

§ 181 Package **marginfix**

(Emulates or patches code by STEPHEN HICKS.)

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*{\parshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][1]{}
10 \newcommand*{\unblockmargin}[1][1]{}
11 \newcommand*{\marginphantom}[2][1]{}

```

File 101 **lwarp-marginnote.sty**

§ 182 Package **marginnote**

(Emulates or patches code by MARKUS KOHM.)

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 102 **lwarp-mcaption.sty**

§ 183 Package **mcaption**

(Emulates or patches code by STEPHAN HENNIG.)

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 103 **lwarp-mdframed.sty**

§ 184 Package **mdframed**

(Emulates or patches code by MARCO DANIEL, ELKE SCHUBERT.)

Pkg mdframed mdframed is loaded with options forced to framemethod=none.

§ 184.1 Package loading

for HTML output:

```
1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackageDrop{mdframed}
```

amsthm must be loaded before mdframed

```
3 \LWR@origRequirePackage{amsthm}
```

Do not require Tikz or pstricks:

```
4 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

§ 184.2 Limitations

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

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

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

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

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

ignored options `userdefinedwidth` and `align` are currently ignored.

CSS classes Environments created or encapsulated by `mdframed` are enclosed in a `<div>` of class `md<environmentname>`, or `mdframed` otherwise.

Frame titles are placed into a `` 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.

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 }

```

§ 184.3 Color and length HTML conversion

`\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 }

```

§ 184.4 Environment encapsulation

`\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@orighspace}%
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 }
```

§ 184.5 Titles and subtitles

`\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  {\sub -or- subsub} [⟨options⟩] {⟨title⟩}

```

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@mdfsubsubtitle [<options>] {<title>}
```

```

119 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
120
121 \newcommand*{\LWR@mdfsubsubtitle}{%
122 \LWR@mdfsubsubtitlecommon{subsub}%

```

```
123 }
124 \let\mdfsubsubtitle\LWR@mdfsubsubtitle
```

§ 184.6 New environments

`\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           \ifstrempy{##1}%
```

```

154     {\let\@temptitle\relax}%
155     {%
156     \def\@temptitle{\mdf@theoremseparator%
157     \mdf@theoremspace%
158     \mdf@theoremtitlefont%
159     ##1}%
160     \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
161     }%
162     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
163     \@temptitle}]]%
164     {\end{mdframed}}%
165     \newenvironment{#2*}[1][1][1]{%
166     \ifstrempy{##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][1][1]{%
177     \refstepcounter{#2}%
178     \ifstrempy{##1}%
179     {\let\@temptitle\relax}%
180     {%
181     \def\@temptitle{\mdf@theoremseparator%
182     \mdf@theoremspace%
183     \mdf@theoremtitlefont%
184     ##1}%
185     \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
186     }
187     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
188     \@temptitle}]]%
189     {\end{mdframed}}%
190     \newenvironment{#2*}[1][1][1]{%
191     \ifstrempy{##1}%
192     {\let\@temptitle\relax}%
193     {%
194     \def\@temptitle{\mdf@theoremseparator%
195     \mdf@theoremspace%
196     \mdf@theoremtitlefont%
197     ##1}%
198     \mdf@thm@caption{#2}{#{4}{\csname the#2\endcsname}{##1}}%
199     }%
200     \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
201     {\end{mdframed}}%
202     }%
203     }%

```

```

204   {%#3 given -- number relationship
205   \global\@namedef{the#2}{\@nameuse{the#3}}%
206   \newenvironment{#2}[1][]{%
207     \refstepcounter{#3}%
208     \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}{#4}{\csname the#2\endcsname}{##1}}%
216     }
217   \begin{mdframed}[#1,frametitle={\strut#4 \csname the#2\endcsname%
218     \@temptitle}]]%
219     {\end{mdframed}}%
220   \newenvironment{#2*}[1][]{%
221     \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{0}{ 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 104 **lwarp-metalogo.sty**

§ 185 Package **metalogo**

(Emulates or patches code by ANDREW GILBERT MOSCHOU.)

Pkg metalogo metalogo is emulated.

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 105 **lwarp-microtype.sty**

§ 186 Package **microtype**

(Emulates or patches code by R SCHLICHT.)

Pkg microtype microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-microtype:

```

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{\slig}{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 106 **lwarp-midfloat.sty**

§ 187 Package **midfloat**

(Emulates or patches code by SIGITAS TOLUŠIS.)

Pkg midfloat midfloat is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{midfloat}

2 \newenvironment{strip}[1] [] {} {}

3 \newskip\stripsep

File 107 **lwarp-moreverb.sty**

§ 188 Package **moreverb**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

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{verbatim}{%
8 \LWR@afterrendverbatim%
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@afterrendverbatim%
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@afterrendverbatim%
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@afterrendverbatim%
39 }

40 \LetLtxMacro\LWRMV@@listinginput\@listinginput
41
42 \renewcommand{\@listinginput}[3][{}]{
43 \LWR@forcenewpage
44 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
45 \LWRMV@@listinginput[#1]{#2}{#3}
46 \LWR@afterrendverbatim%
47 }
48
49
50 \renewenvironment*{boxedverbatim}
51 {
52 \LWR@forcenewpage
```

```

53 \LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-\baselineskip}%
54 \verbatim%
55 }
56 {
57 \endverbatim%
58 \LWR@afterendverbatim%
59 }
60
61

62 \end{warpHTML}

```

File 108 `lwarp-morewrites.sty`

§ 189 Package **morewrites**

Pkg `morewrites` Error if `morewrites` is loaded after `lwarp`.

Discard all options for `lwarp-morewrites`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{morewrites}
2 \LWR@loadbefore{morewrites}

```

File 109 `lwarp-mparhack.sty`

§ 190 Package **mparhack**

Pkg `mparhack` Ignored.

for HTML output: Discard all options for `lwarp-mparhack`:

```

1 \LWR@ProvidesPackageDrop{mparhack}

```

File 110 `lwarp-multicol.sty`

§ 191 Package **multicol**

(Emulates or patches code by FRANK MITTELBACH.)

Pkg `multicol` `multicol` is emulated.

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 * {<numcols>} [<heading>]
```

```
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}{multicolsheading}#3\end{BlockClass}}
```

When done with the environment, close the <div>:

```
8 {\endBlockClass}
```

Emulated null functions which are not used in HTML:

```
9 \newcommand*{\columnbreak}{}
10 \newcommand*{\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 111 **lwarp-multirow.sty**

§ 192 Package **multirow**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg multirow multirow is emulated during HTML output, and used as-is while inside a lateximage.

In a lateximage, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 61.18 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}{}`

§ 192.1 **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}{\arabic{\LWR@tablecolindex}}%
```

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{\arabic{\LWR@tablecolindex}}%
23 "%
```

```
24 \LWR@tdstartstyles%
25 \LWR@addcmidrulewidth%
26 \LWR@tdendstyles%
27 }%
```

The column's < spec:

```
28 \LWR@getexpararray{\LWR@colbeforespec}{\arabic{\LWR@tablecolindex}}%
```

While printing the text, redefine \\\ to generate a new line

```
29 \begingroup\LetLtxMacro{\\}{\LWR@endofline}\#6\endgroup%
30 \LWR@stoppars%
31 \global\boolfalse{\LWR@intabularmetadata}%
32 \renewcommand{\LWR@multirowborder}{}%
33 \LWR@traceinfo{*** multirow done}%
34 }%
```

§ 192.2 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, `\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.

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

```
35 \NewDocumentCommand{\multicolumnrow}{m m O{} m O{} m O{} +m}{%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
36 \LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn/multirow tag:

```
37 \begingroup%
38 \LetLtxMacro{\}\{\LWR@endofline}%
39 \LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
40 \endgroup%
```

Move to the next \LaTeX column:

```
41 \addtocounter{LWR@tablecolindex}{#1}%
42 \addtocounter{LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
43 \booltrue{LWR@skipatbang}%
44 }
```

```

45 \appto{\LWR@restoreorigformatting}{%
46 \LetLtxMacro\multirow\LWR@origmultirow%
47 \renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
48 }

```

File 112 **lwarp-nameref.sty**

§ 193 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 113 **lwarp-needspace.sty**

§ 194 Package **needspace**

(Emulates or patches code by PETER WILSON.)

Pkg **needspace** **needspace** is not used during HTML conversion.

for HTML output: Discard all options for **lwarp-needspace**:

```

1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}

```

File 114 **lwarp-newclude.sty**

§ 195 Package **newclude**

Pkg **newclude** Error if **newclude** is loaded after **lwarp**.

Discard all options for **lwarp-newclude**:

for HTML output: `1 \LWR@ProvidesPackageDrop{newclude}`

```
2 \LWR@loadbefore{newclude}
```

File 115 **lwarp-newunicodechar.sty**

§ 196 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 116 **lwarp-nextpage.sty**

§ 197 Package **nextpage**

(Emulates or patches code by PETER WILSON.)

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 117 **lwarp-nonumonpart.sty**

§ 198 Package **nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

File 118 **lwarp-nopageno.sty**

§ 199 Package **nopageno**

Pkg nopageno nopageno is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

File 119 **lwarp-nowindow.sty**

§ 200 Package **nowindow**

(Emulates or patches code by RAPHAËL PINSON.)

Pkg nowindow nowindow is not used during HTML conversion.

Discard all options for lwarp-nowindow:

for HTML output: 1 \LWR@ProvidesPackageDrop{nowindow}

\nowindow [*<lines>*]
 \setnowindow [*<lines>*]
 2 \newcommand*{\nowindow}[1] [] {}
 3 \newcommand*{\setnowindow}[1] [] {}
 \noclub [*<lines>*]
 \setnoclub [*<lines>*]
 4 \newcommand*{\noclub}[1] [] {}
 5 \newcommand*{\setnoclub}[1] [] {}

File 120 **lwarp-ntheorem.sty**

§ 201 Package **ntheorem**

(Emulates or patches code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg ntheorem ntheorem is patched for use by lwarp.

Table 13: Ntheorem package — CSS styling of theorems and proofs

Theorem: `<div> of class theorembody<theoremstyle>`

Theorem Header: ` of class theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

§ 201.1 Limitations

⚠ **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.

§ 201.2 Options

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ **Disabled options** The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because `lwarp` emulated `hyperref`.

for HTML output: Some disabled options:

```

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{theorem}

```

§ 201.3 Remembering the theorem style

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 setparams@#1\endcsname
48        \@thm{#1}{#1}{#2}
49       }%
50     \global\@namedef{end#1}{\@endtheorem}
51     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\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@addtheorem#1\endcsname{%
77       \noexpand\@num@addtheorem#1{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheorem#1\endcsname{%
79       \noexpand\@nonum@addtheorem#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 }

```

§ 201.4 HTML cross-referencing

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

91 \newcommand{\LWR@inctheorem}{%
92   \addtocounter{LWR@thisautoid}{1}%
93   \LWR@stoppars%
94   \LWR@htmltag{a id="autoid-\arabic{LWR@thisautoid}}{\LWR@htmltag{/a}}%
95   \LWR@startpars%

```

96 }

§ 201.5 `\newtheoremstyle`

The following are patched for css.

These were in individual files `thp.sty` for plain, `thmb.sty` for margin break, etc. They are gathered together here.

Each theorem is encased in a `BlockClass` environment of class `theorembody<style>`.

Each header is encased in an `\InlineClass` of class `theoremheader<style>`.

```

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@intheorem% new
104     #2}%
105     \def\@opargbegintheorem####1####2####3{%
106     \LWR@forcenewpage% new
107     \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108     \LWR@intheorem% new
109     #3}%
110   }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
```

§ 201.6 **Standard styles**

```

114 \renewtheoremstyle{plain}%
115   {\item[\hskip\labelsep \theorem@headerfont
116     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
117   {\item[\hskip\labelsep \theorem@headerfont
118     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
119
120 \renewtheoremstyle{break}%
121   {\item[
122 % \rlap{\vbox{\hbox{
123   \hskip\labelsep \theorem@headerfont
124   \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
125 %   }\hbox{\strut}}}]%
126   ]}%
127   {\item[
128 % \rlap{\vbox{\hbox{
129   \hskip\labelsep \theorem@headerfont
130   \InlineClass{theoremheaderbreak}%
```

```

131         {##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}%
148     {##2\ ##1\theorem@separator}\newline
149 %   }\hbox{\strut}}}
150   ]}%
151   {\item[
152 %   \rlap{\vbox{\hbox{
153     \hskip\labelsep \theorem@headerfont
154     \InlineClass{theoremheaderchangebreak}%
155     {##2\ ##1\ (##3)\theorem@separator}\newline
156 %   }\hbox{\strut}}}
157   ]}
158
159 \renewtheoremstyle{margin}%
160   {\item[\hskip\labelsep\theorem@headerfont
161     \InlineClass{theoremheadermargin}{##2 \qqquad ##1\theorem@separator}
162   ]}%
163   {\item[\hskip\labelsep\theorem@headerfont
164     \InlineClass{theoremheadermargin}{##2 \qqquad ##1\ (##3)\theorem@separator}
165   ]}
166
167 \renewtheoremstyle{marginbreak}%
168   {\item[\hskip\labelsep\theorem@headerfont
169     \InlineClass{theoremheadermarginbreak}%
170     {##2 \qqquad ##1\theorem@separator}\newline
171   ]}%
172   {\item[\hskip\labelsep\theorem@headerfont
173     \InlineClass{theoremheadermarginbreak}%
174     {##2 \qqquad ##1\ (##3)\theorem@separator}\newline
175   ]}
176
177 \renewtheoremstyle{nonumberplain}%
178   {\item[\theorem@headerfont\hskip\labelsep
179     \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
180   {\item[\theorem@headerfont\hskip \labelsep

```

```

181   \InlineClass{theoremheaderplain}{##1\ (##3)\theoremseparator}}
182
183 \renewtheoremstyle{nonumberbreak}%
184   {\item[
185 %   \rlap{\vbox{\hbox{
186     \hskip\labelsep \theoremheaderfont
187     \InlineClass{theoremheaderbreak}{##1\theoremseparator}\newline
188 %   }\hbox{\strut}}}}
189   ]}%
190   {\item[
191 %   \rlap{\vbox{\hbox{
192     \hskip\labelsep \theoremheaderfont
193     \InlineClass{theoremheaderbreak}{##1\ (##3)\theoremseparator}\newline
194 %   }\hbox{\strut}}}}
195   ]}
196
197 \renewtheoremstyle{empty}%
198   {\item[]}%
199   {\item[\theoremheaderfont \hskip\labelsep\relax
200     \InlineClass{theoremheaderplain}{##3}]}
201
202 \renewtheoremstyle{emptybreak}%
203   {\item[]}%
204   {\item[\theoremheaderfont \hskip\labelsep\relax
205     \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

§ 201.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

206 \ifbool{LWR@theoremamsthm}{}{%
207 % upright text via CSS
208   \newtheoremstyle{plainupright}%
209   {\item[\hskip\labelsep \theoremheaderfont
210     \InlineClass{theoremheaderplain}{##1\ ##2\theoremseparator}}}%
211   {\item[\hskip\labelsep \theoremheaderfont
212     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theoremseparator}}]}
213
214 % upright text and small caps header via CSS
215   \newtheoremstyle{nonumberplainuprightsc}%
216   {\item[\theoremheaderfont\hskip\labelsep
217     \InlineClass{theoremheadersc}{##1\theoremseparator}}}%
218   {\item[\theoremheaderfont\hskip \labelsep
219     \InlineClass{theoremheadersc}{##1\ (##3)\theoremseparator}}]}
220}% not amsthm

```

§ 201.8 Renewed standard configuration

The following standard configuration is renewed using the new css:

```

221 \ifbool{LWR@ntheoremamsthm}{}{%
222   \theoremstyle{plainupright}
223   \theorembodyfont{\upshape}
224   \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
225   \renewtheorem{Example}{Example}
226   \renewtheorem{example}{Example}
227   \renewtheorem{Beispiel}{Beispiel}
228   \renewtheorem{beispiel}{Beispiel}
229   \renewtheorem{Bemerkung}{Bemerkung}
230   \renewtheorem{bemerkung}{Bemerkung}
231   \renewtheorem{Anmerkung}{Anmerkung}
232   \renewtheorem{anmerkung}{Anmerkung}
233   \renewtheorem{Remark}{Remark}
234   \renewtheorem{remark}{Remark}
235   \renewtheorem{Definition}{Definition}
236   \renewtheorem{definition}{Definition}
237
238   \theoremstyle{nonumberplainuprightsc}
239   \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
240   \renewtheorem{Proof}{Proof}
241   \renewtheorem{proof}{Proof}
242   \renewtheorem{Beweis}{Beweis}
243   \renewtheorem{beweis}{Beweis}
244   \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
245
246   \theoremsymbol{}
247 }% not amsthm

```

§ 201.9 amsthm option

Only if the amsthm option was given:

```

248 \ifbool{LWR@ntheoremamsthm}{
249
250 \gdef\th@plain{%
251   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
252   \def\@begintheorem##1##2{%
253 \LWR@forcenewpage% new
254     \BlockClass{theorembodyplain}% new
255     \LWR@inctheorem% new
256     \item[\hskip\labelsep
257 % \theorem@headerfont
258 \InlineClass{theoremheaderplain}{##1\ ##2.}
259   ]}%
260   \def\@opargbegintheorem##1##2##3{%
261 \LWR@forcenewpage% new

```

```

262     \BlockClass{theorembodyplain}% new
263     \LWR@inctheorem% new
264     \item[\hskip\labelsep
265 % \theorem@headerfont
266 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
267     ]}}
268
269 \gdef\th@nonumberplain{%
270 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
271 \def\@begintheorem##1##2{%
272 \LWR@forcenewpage% new
273     \BlockClass{theorembodyplain}% new
274     \LWR@inctheorem% new
275     \item[\hskip\labelsep
276 % \theorem@headerfont
277 \InlineClass{theoremheaderplain}{##1.}
278     ]}%
279 \def\@opargbegintheorem##1##2##3{%
280 \LWR@forcenewpage% new
281     \BlockClass{theorembodyplain}% new
282     \LWR@inctheorem% new
283     \item[\hskip\labelsep
284 % \theorem@headerfont
285 \InlineClass{theoremheaderplain}{##1\ (##3).}
286     ]}}
287
288 \gdef\th@definition{%
289 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
290 \def\@begintheorem##1##2{%
291 \LWR@forcenewpage% new
292     \BlockClass{theorembodydefinition}% new
293     \LWR@inctheorem% new
294     \item[\hskip\labelsep
295 % \theorem@headerfont
296 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
297     ]}%
298 \def\@opargbegintheorem##1##2##3{%
299 \LWR@forcenewpage% new
300     \BlockClass{theorembodydefinition}% new
301     \LWR@inctheorem% new
302     \item[\hskip\labelsep
303 % \theorem@headerfont
304 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
305     ]}}
306
307 \gdef\th@nonumberdefinition{%
308 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
309 \def\@begintheorem##1##2{%
310 \LWR@forcenewpage% new
311     \BlockClass{theorembodydefinition}% new

```

```

312         \LWR@inctheorem% new
313         \item[\hskip\labelsep
314% \theorem@headerfont
315 \InlineClass{theoremheaderdefinition}{##1.}
316     ]}%
317 \def\@opargbegintheorem##1##2##3{%
318 \LWR@forcenewpage% new
319     \BlockClass{theorembodydefinition}% new
320     \LWR@inctheorem% new
321     \item[\hskip\labelsep
322% \theorem@headerfont
323 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
324     ]}}
325
326 \gdef\th@remark{%
327 \def\theorem@headerfont{\itshape}\normalfont%
328 \def\@begintheorem##1##2{%
329 \LWR@forcenewpage% new
330     \BlockClass{theorembodyremark}% new
331     \LWR@inctheorem% new
332     \item[\hskip\labelsep
333% \theorem@headerfont
334 \InlineClass{theoremheaderremark}{##1\ ##2.}
335     ]}%
336 \def\@opargbegintheorem##1##2##3{%
337 \LWR@forcenewpage% new
338     \BlockClass{theorembodyremark}% new
339     \LWR@inctheorem% new
340     \item[\hskip\labelsep
341% \theorem@headerfont
342 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
343     ]}}
344
345 \gdef\th@nonumberremark{%
346 \def\theorem@headerfont{\itshape}\normalfont%
347 \def\@begintheorem##1##2{%
348 \LWR@forcenewpage% new
349     \BlockClass{theorembodyremark}% new
350     \LWR@inctheorem% new
351     \item[\hskip\labelsep
352% \theorem@headerfont
353 \InlineClass{theoremheaderremark}{##1.}
354     ]}%
355 \def\@opargbegintheorem##1##2##3{%
356 \LWR@forcenewpage% new
357     \BlockClass{theorembodyremark}% new
358     \LWR@inctheorem% new
359     \item[\hskip\labelsep
360% \theorem@headerfont
361 \InlineClass{theoremheaderremark}{##1\ (##3).}

```

```
362     ]}}
363
364 \gdef\th@proof{%
365   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
366   \def\@begintheorem##1##2{%
367     \LWR@forcenewpage% new
368       \BlockClass{theorembodyproof}% new
369       \LWR@inctheorem% new
370       \item[\hskip\labelsep
371 % \theorem@headerfont
372 \InlineClass{theoremheaderproof}{##1.}
373   ]}%
374   \def\@opargbegintheorem##1##2##3{%
375     \LWR@forcenewpage% new
376       \BlockClass{theorembodyproof}% new
377       \LWR@inctheorem% new
378       \item[\hskip\labelsep
379 % \theorem@headerfont
380 \InlineClass{theoremheaderproof}{##1\ (#3).}
381   ]}}
382
383
384
385 \newcounter{proof}%
386 \if@thmmarks
387   \newcounter{currproofctr}%
388   \newcounter{endproofctr}%
389 \fi
390
391 \gdef\proofSymbol{\openbox}
392
393 \newcommand{\proofname}{Proof}
394
395 \newenvironment{proof}[1][\proofname]{
396   \th@proof
397   \def\theorem@headerfont{\itshape}%
398   \normalfont
399   \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
400   \@thm{proof}{proof}{#1}
401 }%
402 {\@endtheorem}
403
404 }{}% amsthm option
```

§ 201.10 **Ending a theorem**

Patched for CSS:

```

405 \let\LWR@origendtheorem\endtheorem
406 \renewcommand{\@endtheorem}{%
407 \ifbool{LWR@theoremmarks}{%
408   \ifsetendmark%
409   \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
410   \setendmarkfalse%
411   \fi%
412 }{}%
413 \LWR@origendtheorem%
414 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
415 \endBlockClass%
416 }

```

§ 201.11 **\NoEndMark**

```

417 \gdef\NoEndMark{\global\setendmarkfalse}

```

§ 201.12 **List-of**

Redefined to reuse the float mechanism to add list-of-theorem links:

```

\thm@thmline {<1: printed type>} {<2: #>} {<3: optional>} {<4: page>}

```

```

418 \renewcommand{\thm@thmline@noname}[4]{%
419 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
420 }
421
422 \renewcommand{\thm@thmline@name}[4]{%
423 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
424 }

```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```

425 \def\thm@thmline{\thm@thmline@name}

```

Patch for CSS:

```

426 \def\listtheorems#1{
427 \LWR@htmlclass{nav}{lothm}%
428 \begingroup
429 \c@tocdepth=-2%
430 \def\thm@list{#1}\thm@processlist
431 \endgroup
432 \LWR@htmlclassend{nav}{lothm}%
433 }

```

§ 201.13 Symbols

Proof QED symbol:

```
434 \newcommand{\qed}{\quad\the\qedsymbol}
435
436 \AtBeginDocument{
437 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
438 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
439 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
440 }
```

§ 201.14 Cross-referencing

`\thref {<label>}`

```
441 \newcommand*{\thref}[1]{\cref{#1}}
```

File 121 lwarp-overpic.sty

§ 202 Package **overpic**

(Emulates or patches code by ROLF NIEPRASCHK.)

Pkg overpic overpic is patched for use by lwarp.

 **scaling** The macros `\overpicfontsize` and `\overpicfontskip` are used during HTML generation. These are sent to `\fontsize` to adjust the font size for scaling differences between the print and HTML versions of the document. Renew these macros before using the `overpic` and `Overpic` environments.

See section [72.2](#) for the print-mode version of `\overpicfontsize` and `\overpicfontskip`.

for HTML output:

```
1 \LWR@ProvidesPackagePass{overpic}
2
3 \newcommand*{\overpicfontsize}{12}
4 \newcommand*{\overpicfontskip}{14}
5
6 \BeforeBeginEnvironment{overpic}{%
7   \begin{lateximage}%
8   \fontsize{\overpicfontsize}{\overpicfontskip}%
9   \selectfont%
10 }
11 \AfterEndEnvironment{overpic}{\end{lateximage}}
12
13 \BeforeBeginEnvironment{Overpic}{%
```

```

14 \begin{lateximage}%
15 \fontsize{\overpicfontsize}{\overpicfontskip}%
16 \selectfont%
17 }
18
19 \AfterEndEnvironment{Overpic}{\end{lateximage}}

```

File 122 **lwarp-pagenote.sty**

§ 203 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 123 **lwarp-paralist.sty**

§ 204 Package **paralist**

(Emulates or patches code by BERND SCHANDL.)

Pkg paralist paralist is supported with minor changes.

for HTML output:

```

1 \LWR@ProvidesPackagePass{paralist}

```

The compact environments are identical to the regular ones:

```

2 \AtBeginEnvironment{compactitem}{\LWR@itemizestart}
3 \AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
4 \AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
5 \AtEndEnvironment{compactitem}{\LWR@listend}
6 \AtEndEnvironment{compactenum}{\LWR@listend}
7 \AtEndEnvironment{compactdesc}{\LWR@listend}

```

For the inline environments, revert `\item` to its original print-mode version:

```

8 \AtBeginEnvironment{inparaitem}{\LetLtxMacro\item\LWR@origitem}
9 \AtBeginEnvironment{inparaenum}{\LetLtxMacro\item\LWR@origitem}
10 \AtBeginEnvironment{inparadesc}{\LetLtxMacro\item\LWR@origitem}

```

Manual formatting of the description labels:

```

11 \def\paradescriptionlabel#1{\normalfont\textbf{#1}}

```

File 124 **lwarp-parskip.sty**

§ 205 Package **parskip**

Pkg `parskip` `parskip` is ignored.

for HTML output: Discard all options for `lwarp-parskip`.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

File 125 **lwarp-pdfscape.sty**

§ 206 Package **pdfscape**

Pkg `pdfscape` `Emulated`.

for HTML output: Discard all options for `lwarp-pdfscape`:

```
1 \LWR@ProvidesPackageDrop{pdfscape}
```

File 126 **lwarp-pdfsync.sty**

§ 207 Package **pdfsync**

(Emulates or patches code by J. LAURENS.)

Pkg `pdfsync` `Emulated`.

for HTML output: Discard all options for `lwarp-pdfsync`:

```
1 \LWR@ProvidesPackageDrop{pdfsync}
```

```
2 \newcommand*{\pdfsync}{}
3 \newcommand*{\pdfsyncstart}{}
4 \newcommand*{\pdfsyncstop}{}

```

File 130 **lwarp-preview.sty**

§ 211 Package **preview**

Pkg preview preview is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{preview}

2 \newenvironment{preview}{}{}
3 \newenvironment{nopreview}{}{}
4 \NewDocumentCommand{\PreviewMacro}{s o o +m}{}
5 \NewDocumentCommand{\PreviewEnvironment}{s o o +m}{}
6 \newcommand{\PreviewSnarfEnvironment}[2] [] {}
7 \NewDocumentCommand{\PreviewOpen}{s o}{}
8 \NewDocumentCommand{\PreviewClose}{s o}{}
9 \let\ifPreview\iffalse% \fi for syntax highlighting

```

File 131 **lwarp-quotchap.sty**

§ 212 Package **quotchap**

(Emulates or patches code by KARSTEN TINNEFELD, JAN KLEVER.)

Pkg quotchap quotchap is emulated.

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 132 **lwarp-ragged2e.sty**

§ 213 Package **ragged2e**

(Emulates or patches code by MARTIN SCHRÖDER.)

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

for HTML output:

```
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 133 **lwarp-realscripts.sty**

§ 214 Package **realscripts**

(Emulates or patches code by WILL ROBERTSON.)

Pkg realscripts realscripts is emulated. See lwarp.css for the of class supsubscript.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{realscripts}

2 \let\realsuperscript\textsuperscript
3 \let\realsubscript\textsubscript
4
5 \let\fakesuperscript\textsuperscript
6 \let\fakesubscript\textsubscript
7
8 \newlength{\subsupersep}
9
10 \newcommand*{\LWR@realscriptsalign}{}
11
12 \newcommand*{\LWR@setrealscriptsalign}[1]{%
13 \renewcommand*{\LWR@realscriptsalign}{}%
14 \ifthenelse{\equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{text-align:center;}}{}%
15 \ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{text-align:right;}}{}%
16 }
17
18 \DeclareDocumentCommand \textsubsuperscript {s 0{1} mm} {%
19 \LWR@setrealscriptsalign{#2}%
20 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%

```

```

21 \textsuperscript{#4}\textsubscript{#3}%
22 }%
23 }
24
25 \DeclareDocumentCommand \textsupersubscript {s O{1} mm} {%
26 \LWR@setrealscriptsalign{#2}%
27 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
28 \textsubscript{#4}\textsuperscript{#3}%
29 }%
30 }

```

File 134 **lwarp-reysize.sty**

§ 215 Package **relsize**

(Emulates or patches code by DONALD ARSENEAU, BERNIE COSELL, MATT SWIFT.)

Pkg **relsize** relsize is patched for use by lwarp.

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

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

While creating SVG math for HTML, the original definitions are 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@relesizetemp}
21
22 \renewcommand*\relesize}[1]{}
23 \renewcommand*\larger}[1] []{}
24 \renewcommand*\smaller}[1] []{}
25 \renewcommand*\relescale}[1]{}
26
27 \renewcommand*\textlarger}[2][1]{%
28 \setcounter{LWR@relesizetemp}{100+(#1*20)}%
29 \InlineClass[font-size:\arabic{LWR@relesizetemp}\%]{textlarger}{#2}%
30 }
31
32 \renewcommand*\textsmaller}[2][1]{%
33 \setcounter{LWR@relesizetemp}{100-(#1*20)}%
34 \InlineClass[font-size:\arabic{LWR@relesizetemp}\%]{textsmaller}{#2}%
35 }
36
37 \renewcommand*\textscale}[2]{%
38 \setcounter{LWR@relesizetemp}{100*\real{#1}}%
39 \InlineClass[font-size:\arabic{LWR@relesizetemp}\%]{textscale}{#2}%
40 }

```

File 135 **lwarp-romanbar.sty**

§ 216 Package **romanbar**

(Emulates or patches code by H.-MARTIN MÜNCH.)

Pkg romanbar romanbar is patched for use by lwarp.

An inline class with an overline and underline is used.

for HTML output:

```

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 136 **lwarp-romanbarpagenumber.sty**

§ 217 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{romanbarpagenumber}

File 137 **lwarp-rotating.sty**

§ 218 Package **rotating**

(Emulates or patches code by ROBIN FAIRBAIRNS, SEBASTIAN RAHTZ, LEONOR BARROCA.)

Pkg rotating rotating is emulated.

All rotations are ignored in HTML output.

for HTML output: 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 138 **lwarp-rotfloat.sty**

§ 219 Package **rotfloat**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg rotfloat rotfloat is emulated.

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 139 `lwarp-savetrees.sty`

§ 220 Package **savetrees**

Pkg `savetrees` Emulated.

for HTML output: Discard all options for `lwarp-savetrees`:

```

1 \LWR@ProvidesPackageDrop{savetrees}

```

File 140 **lwarp-scalefnt.sty**

§ 221 Package **scalefnt**

(Emulates or patches code by D. CARLISLE.)

Pkg `scalefnt` `scalefnt` is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{scalefnt}

2 \DeclareRobustCommand\scalefont[1]{}

```

File 141 **lwarp-scrextend.sty**

§ 222 Package **scrextend**

Pkg `scrextend` `scrextend` is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{scrextend}

2 \DeclareDocumentCommand{\setkomafont}{m m}{}
3 \DeclareDocumentCommand{\addkomafont}{m m}{}
4 \DeclareDocumentCommand{\usekomafont}{m}{}
5
6 \DeclareDocumentCommand{\usefontofkomafont}{m}{}
7 \DeclareDocumentCommand{\useencodingofkomafont}{m}{}
8 \DeclareDocumentCommand{\usesizeofkomafont}{m}{}
9 \DeclareDocumentCommand{\usefamilyofkomafont}{m}{}
10 \DeclareDocumentCommand{\useseriesofkomafont}{m}{}
11 \DeclareDocumentCommand{\useshapeofkomafont}{m}{}
12
13 \AtBeginDocument{
14 \let\LWR@maketitle\maketitle
15 \DeclareDocumentCommand{\maketitle}{o}{\LWR@maketitle}
16 }
17
18 \DeclareDocumentCommand{\extratitle}{m}{}
19 \DeclareDocumentCommand{\titlehead}{m}{}

```

```

20 \DeclareDocumentCommand{\subject}{m}{}
21 \DeclareDocumentCommand{\publishers}{m}{\published{#1}}
22 \DeclareDocumentCommand{\uppertitleback}{m}{}
23 \DeclareDocumentCommand{\lowertitleback}{m}{}
24 \DeclareDocumentCommand{\dedication}{m}{}
25
26 \DeclareDocumentCommand{\ifthispageodd}{m m}{#1}
27
28 \DeclareDocumentCommand{\titlepagestyle}{}{}
29
30 \DeclareDocumentCommand{\cleardoublepageusingstyle}{m}{}
31 \DeclareDocumentCommand{\cleardoubleemptypage}{}{}
32 \DeclareDocumentCommand{\cleardoubleplainpage}{}{}
33 \DeclareDocumentCommand{\cleardoublestandardpage}{}{}
34 \DeclareDocumentCommand{\cleardoubleoddpge}{}{}
35 \DeclareDocumentCommand{\cleardoubleoddpgeusingstyle}{m}{}
36 \DeclareDocumentCommand{\cleardoubleoddpgeemptypage}{}{}
37 \DeclareDocumentCommand{\cleardoubleoddpgeplainpage}{}{}
38 \DeclareDocumentCommand{\cleardoubleoddpgestandardpage}{}{}
39 \DeclareDocumentCommand{\cleardoubleevenpage}{}{}
40 \DeclareDocumentCommand{\cleardoubleevenpageusingstyle}{m}{}
41 \DeclareDocumentCommand{\cleardoubleevenemptypage}{}{}
42 \DeclareDocumentCommand{\cleardoubleevenplainpage}{}{}
43 \DeclareDocumentCommand{\cleardoubleevenstandardpage}{}{}
44
45 \DeclareDocumentCommand{\multiplefootnoteseparator}{}{%
46   \begingroup\let\thefootnotemark\multfootsep\makefnmark\endgroup
47 }
48
49 \DeclareDocumentCommand{\multfootsep}{}{,}
50
51 \DeclareDocumentCommand{\footref}{m}{%
52   \begingroup
53     \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
54   \endgroup
55   \@footnotemark
56 }
57
58 \DeclareDocumentCommand{\deffootnote}{o m m m}{}
59 \DeclareDocumentCommand{\deffootnotemark}{m}{}
60 \DeclareDocumentCommand{\setfootnoterule}{o m}{}
61 \DeclareDocumentCommand{\raggedfootnote}{}{}
62
63 \DeclareDocumentCommand{\dictum}{o m}{
64 \begin{LWR@BlockClassWP}{text-align:right}{}{dictum}
65   #2
66   \IfValueT{#1}
67   {
68     \ifbool{FormatWP}
69     {\begin{BlockClass}[border-top:1px solid gray]{dictumauthor}}

```

```

70     {\begin{BlockClass}{dictumauthor}}
71     \dictumauthorformat{#1}
72     \end{BlockClass}
73   }
74 \end{LWR@BlockClassWP}
75 }
76
77 \DeclareDocumentCommand{\dictumwidth}{-}{-}
78 \DeclareDocumentCommand{\dictumauthorformat}{m}{(#1)}
79 \DeclareDocumentCommand{\dictumrule}{-}{-}
80 \DeclareDocumentCommand{\raggeddictum}{-}{-}
81 \DeclareDocumentCommand{\raggeddictumtext}{-}{-}
82 \DeclareDocumentCommand{\raggeddictumauthor}{-}{-}
83
84 \DeclareDocumentEnvironment{labeling}{o m}
85 {%
86 \def\sc@septext{#1}%
87 \list{}{}%
88 \let\makelabel\labelinglabel%
89 }
90 {
91 \endlist
92 }
93
94 \DeclareDocumentCommand{\labelinglabel}{m}{%
95 #1 \qqquad \sc@septext%
96 }
97
98 \let\addmargin\relax
99 \let\endaddmargin\relax
100 \cslet{addmargin*}{\relax}
101 \cslet{endaddmargin*}{\relax}
102
103 \NewDocumentEnvironment{addmargin}{s O{} m}
104 {
105 \uselengthunit{PT}%
106 \setlength{\LWR@templengthtwo}{#3}
107 \ifblank{#2}
108 {
109   \begin{BlockClass}[
110     margin-left:\rndprintlength{\LWR@templengthtwo} ;
111     margin-right:\rndprintlength{\LWR@templengthtwo}
112   ]{addmargin}
113 }
114 {
115   \setlength{\LWR@templengthone}{#2}
116   \begin{BlockClass}[
117     margin-left:\rndprintlength{\LWR@templengthone} ;
118     margin-right:\rndprintlength{\LWR@templengthtwo}
119   ]{addmargin}

```

```

120 }
121 }
122 {\end{BlockClass}}

```

Ref to create a starred environment:

<https://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>

```

123
124 \ExplSyntaxOn
125 \cs_new:cpn {addmargin*} {\addmargin*}
126 \cs_new_eq:cN {endaddmargin*} \endaddmargin
127 \ExplSyntaxOff
128
129 \DeclareDocumentCommand{\marginline}{m}{\marginpar{#1}}

```

File 142 **lwarp-scrhack.sty**

§ 223 Package **scrhack**

Pkg scrhack scrhack is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{scrhack}

File 143 **lwarp-sclayer.sty**

§ 224 Package **sclayer**

(Emulates or patches code by MARKUS KOHM.)

Pkg sclayer sclayer is emulated.

 **Not yet tested!** Please send bug reports!

for HTML output: 1 \LWR@ProvidesPackageDrop{sclayer}

```

2 \newcommand*{\DeclareSectionNumberDepth}[2]{}
3 \newcommand*{\DeclareLayer}[2] [] {}
4 \newcommand*{\DeclareNewLayer}[2] [] {}
5 \newcommand*{\ProvideLayer}[2] [] {}
6 \newcommand*{\RedeclareLayer}[2] [] {}
7 \newcommand*{\ModifyLayer}[2] [] {}
8 \newcommand*{\layerhalign}{}
9 \newcommand*{\layervalign}{}

```

```
10 \newcommand*\layerxoffset{}
11 \newcommand*\laveryoffset{}
12 \newcommand*\layerwidth{}
13 \newcommand*\layerheight{}
14 \providecommand*\LenToUnit[1]{\strip@pt\dimexpr#1*\p@/\unitlength}
15 \newcommand*\putUL[1]{}
16 \newcommand*\putUR[1]{}
17 \newcommand*\putLL[1]{}
18 \newcommand*\putLR[1]{}
19 \newcommand*\putC[1]{}
20 \newcommand*\GetLayerContents[1]{}
21 \newcommand*\IfLayerExists[3]{#3}
22 \newcommand*\DestroyLayer[1]{}
23 \newcommand*\layercontentsmeasure{}
24 \newcommand*\currentpagestyle{}
25 \newcommand*\BeforeSelectAnyPageStyle[1]{}
26 \newcommand*\AfterSelectAnyPageStyle[1]{}
27 \newcommand*\DeclarePageStyleAlias[2]{}
28 \newcommand*\DeclareNewPageStyleAlias[2]{}
29 \newcommand*\ProvidePageStyleAlias[2]{}
30 \newcommand*\RedeclarePageStyleAlias[2]{}
31 \newcommand*\DestroyPageStyleAlias[1]{}
32 \newcommand*\GetRealPageStyle[1]{}
33 \newcommand*\DeclarePageStyleByLayers[3]{}
34 \newcommand*\DeclareNewPageStyleByLayers[3]{}
35 \newcommand*\ProvidePageStyleByLayers[3]{}
36 \newcommand*\RedeclarePageStyleByLayers[3]{}
37 \NewDocumentCommand*\ForEachLayerOfPageStyle{s m m}{}
38 \newcommand*\AddLayersToPageStyle[2]{}
39 \newcommand*\AddLayersAtBeginOfPageStyle[2]{}
40 \newcommand*\AddLayersAtEndOfPageStyle[2]{}
41 \newcommand*\RemoveLayersFromPageStyle[2]{}
42 \newcommand*\AddLayersToPageStyleBeforeLayer[3]{}
43 \newcommand*\AddLayersToPageStyleAfterLayer[3]{}
44 \newcommand*\UnifyLayersAtPageStyle[1]{}
45 \newcommand*\ModifyLayerPageStyleOptions[2]{}
46 \newcommand*\AddToLayerPageStyleOptions[2]{}
47 \newcommand*\IfLayerPageStyleExists[3]{#3}
48 \newcommand*\IfRealLayerPageStyleExists[3]{#3}
49 \newcommand*\IfLayerAtPageStyle[4]{#4}
50 \newcommand*\IfSomeLayerAtPageStyle[4]{#4}
51 \newcommand*\IfLayersAtPageStyle[4]{#4}
52 \newcommand*\DestroyRealLayerPageStyle[1]{}
53 \@ifundefined{footheight}{\newlength\footheight}{}
54 \DeclareDocumentCommand*\automark{s o m}{}
55 \DeclareDocumentCommand*\manualmark{}{}
56 \DeclareDocumentCommand*\MakeMarkcase{m}{#1}
57 \DeclareDocumentCommand*\GenericMarkFormat{}{}
58 \newcommand*\@mkleft[1]{}
59 \newcommand*\@mkright[1]{}

```

```

60 \newcommand*{\@mkdouble}[1]{}
61 \newcommand*{\@mkboth}[2]{}
62 \newcommand*{\sclayerInitInterface}[1] [] {}
63 \newcommand*{\sclayerAddToInterface}[3] [] {}
64 \newcommand*{\sclayerAddCsToInterface}[3] [] {}
65 \newcommand*{\sclayerOnAutoRemoveInterface}[2] [] {}

```

File 144 **lwarp-sclayer-notecolumn.sty**

§ 225 Package **sclayer-notecolumn**

(Emulates or patches code by MARKUS KOHM.)

Pkg sclayer-notecolumn sclayer-notecolumn is emulated.

 **Not yet tested!** [Please send bug reports!](#)

for HTML output:

```

1 \LWR@ProvidesPackageDrop{sclayer-notecolumn}

2 \newcommand*{\DeclareNoteColumn}[2] [] {}
3 \newcommand*{\DeclareNewNoteColumn}[2] [] {}
4 \newcommand*{\ProvideNoteColumn}[2] [] {}
5 \newcommand*{\RedeclareNoteColumn}[2] [] {}
6 \NewDocumentCommand{\makenote}{s o m}{\marginpar{#3}}
7 \newcommand*{\syncwithnotecolumn}[1] [] {}
8 \newcommand*{\syncwithnotecolumns}[1] [] {}
9 \newcommand*{\clearnotecolumn}[1] [] {}
10 \newcommand*{\clearnotecolumns}[1] [] {}

```

File 145 **lwarp-sclayer-scrpage.sty**

§ 226 Package **sclayer-scrpage**

(Emulates or patches code by MARKUS KOHM.)

Pkg sclayer-scrpage sclayer-scrpage is emulated.

 **Not yet tested!** [Please send bug reports!](#)

for HTML output:

```

1 \LWR@ProvidesPackageDrop{sclayer-scrpage}

2 \@ifundefined{footheight}{\newlength{footheight}}{}
3 \NewDocumentCommand{\lehead}{s o m}{}
4 \NewDocumentCommand{\cehead}{s o m}{}

```

```

5 \NewDocumentCommand{\rehead}{s o m}{}
6 \NewDocumentCommand{\lohead}{s o m}{}
7 \NewDocumentCommand{\cohead}{s o m}{}
8 \NewDocumentCommand{\rohead}{s o m}{}
9 \NewDocumentCommand{\lefoot}{s o m}{}
10 \NewDocumentCommand{\cefoot}{s o m}{}
11 \NewDocumentCommand{\refoot}{s o m}{}
12 \NewDocumentCommand{\lofoot}{s o m}{}
13 \NewDocumentCommand{\cofoot}{s o m}{}
14 \NewDocumentCommand{\rofoot}{s o m}{}
15 \NewDocumentCommand{\ohead}{s o m}{}
16 \NewDocumentCommand{\chead}{s o m}{}
17 \NewDocumentCommand{\ihead}{s o m}{}
18 \NewDocumentCommand{\ofoot}{s o m}{}
19 \NewDocumentCommand{\cfoot}{s o m}{}
20 \NewDocumentCommand{\ifoot}{s o m}{}
21 \DeclareDocumentCommand{\automark}{s o m}{}
22 \DeclareDocumentCommand{\manualmark}{}{}
23 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
24 \DeclareDocumentCommand{\GenericMarkFormat}{}{}
25 \newcommand*{\defpairofpagestyles}[3] [] {}
26 \newcommand*{\newpairofpagestyles}[3] [] {}
27 \newcommand*{\renewpairofpagestyles}[3] [] {}
28 \newcommand*{\providepairofpagestyles}[3] [] {}
29 \newcommand*{\clearmainofpairofpagestyles}
30 \newcommand*{\clearplainofpairofpagestyles}
31 \newcommand*{\clearpairofpagestyles}
32 \NewDocumentCommand{\deftriplepagestyle}{m o o m m m m m m}{}
33 \NewDocumentCommand{\newtriplepagestyle}{m o o m m m m m m}{}
34 \NewDocumentCommand{\renewtriplepagestyle}{m o o m m m m m m}{}
35 \NewDocumentCommand{\providetriplepagestyle}{m o o m m m m m m}{}
36 \newcommand*{\defpagestyle}[3] {}
37 \newcommand*{\newpagestyle}[3] {}
38 \newcommand*{\providepagestyle}[3] {}
39 \newcommand*{\renewpagestyle}[3] {}

```

File 146 **lwarp-section.sty**

§ 227 Package **section**

Pkg section section is ignored.

(Emulates or patches code by OLIVER PRETZEL.)

for HTML output: 1 \LWR@ProvidesPackageDrop{section}

2 \ifx\chapter\undefined

```

3 \def\chsize{\Large}\def\hdsi{size}{\huge}\else
4 \def\chsize{\huge}\def\hdsi{size}{\Huge}
5 \fi
6 \let\ttsi{size}\LARGE
7 \let\ausi{size}\large
8 \let\dasi{size}\large
9 \let\secsi{size}\Large
10 \let\subsi{size}\large
11 \let\hdpos\raggedright
12 \newcount{hddepth}
13 \let\fpind\relax
14 \def\ttfnt{}
15 \def\hdfnt{}
16 \def\fefnt{}
17 \def\thfnt{}
18 \def\pgfnt{}
19 \def\hmkfnt{}
20 \let\mkcse\uppercase
21 \def\hddot{}
22 \def\cpdot{:}
23 \def\nm{dot}{}
24 \ifx\secindent\undefined
25 \newdimen\secindent
26 \newskip\secpreskip
27 \newskip\secpstskip
28 \newdimen\subindent
29 \newskip\subpreskip
30 \newskip\subpstskip
31 \newskip\parpstskip
32 \newcount\c@hddepth
33 \fi

```

File 147 **lwarp-sectsty.sty**

§ 228 Package **sectsty**

(Emulates or patches code by ROWLAND MCDONNELL.)

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 148 **lwarp-setspace.sty**

§ 229 Package **setspace**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg `setspace` `setspace` is not used during HTML conversion.

Discard all options for `lwarp-setspace`:

for HTML output:

```

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 149 **lwarp-shadow.sty**

§ 230 Package **shadow**

(Emulates or patches code by MAURO ORLANDINI.)

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 150 **lwarp-showidx.sty**

§ 231 Package **showidx**

Pkg `showidx` `showidx` is ignored.

for HTML output: Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

File 151 **lwarp-showkeys.sty**

§ 232 Package **showkeys**

(Emulates or patches code by DAVID CARLISLE, MORTEN HØGHOLM.)

Pkg `showkeys` `showkeys` is ignored.

for HTML output: Discard all options for `lwarp-showkeys`:

```
1 \LWR@ProvidesPackageDrop{showkeys}
2 \NewDocumentCommand{\showkeys}{s}{}

```

File 152 **lwarp-sidecap.sty**

§ 233 Package **sidecap**

(Emulates or patches code by ROLF NIEPRASCHK, HUBERT GÄSSLEIN.)

Pkg `sidecap` `sidecap` is emulated.

for HTML output: Discard all options for `lwarp-sidecap`.

```
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 153 **lwarp-sidenotes.sty**

§ 234 Package **sidenotes**

(Emulates or patches 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{KFLTsidenotes@marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{-}{marginblock}%
25 \captionsetup{type=#2}%
26 }
27 {%
28 \endLWR@BlockClassWP%
29 }
30
31 \RenewDocumentEnvironment{marginfigure}{o}
32   {\begin{KFLTsidenotes@marginfloat}{figure}}
33   {\end{KFLTsidenotes@marginfloat}}
34
35 \RenewDocumentEnvironment{margintable}{o}
36   {\begin{KFLTsidenotes@marginfloat}{table}}
37   {\end{KFLTsidenotes@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the $\LaTeX 2_{\epsilon}$ `article.cls` source:

```

38 \renewenvironment{figure*}
39   {\@dblfloat{figure}}
40   {\end@dblfloat}
41
42 \renewenvironment{table*}
43   {\@dblfloat{table}}
44   {\end@dblfloat}

```

File 154 **lwarp-siunitx.sty**

§ 235 Package **siunitx**

(Emulates or patches code by JOSEPH WRIGHT.)

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.

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

NOTE: As of this writing, the `siunitx` extension for MathJax is not currently hosted at any public CDN, thus `siunitx` is not usable with MathJax unless a local copy of this extension is created first.

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
```

The following is executed by `siunitx` at the end of `\document`. `\@ensuredmath` is not supported inside an `\hbox`, so it must temporarily be restored to its original.

```
9 \ExplSyntaxOn
10 \cs_undefine:N \__siunitx_set_math_fam:n
11 \cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
12   \LWR@traceinfo{siunitx set math fam}
13   \int_new:c { c__siunitx_math #1 _int }
14   \group_begin:
15     \LetLtxMacro\@ensuredmath\LWR@origensuredmath
16     \hbox_set:Nn \l__siunitx_tmp_box
17       {
18         \ensuremath
19           {
20             \use:c { math #1 }
21             {
22               \int_gset:cn { c__siunitx_math #1 _int } { \fam }
23             }
24           }
25       }
26 \ExplSyntaxOff
```

```

24     }
25   }
26 \group_end:
27   \LWR@traceinfo{sunitx set math fam: done}
28 }
29
30 \cs_undefine:N \__siunitx_combined_output:n
31 \cs_new_protected:Npn \__siunitx_combined_output:n #1 {
32   \group_begin:
33   \LetLtxMacro\@ensuredmath\LWR@origensuredmath
34   \bool_if:NTF \l__siunitx_number_parse_bool
35   {
36     \tl_clear:N \l__siunitx_number_out_tl
37     \bool_set_false:N \l__siunitx_number_compound_bool
38     \__siunitx_number_output_parse:n {#1}
39   }
40   {
41     \__siunitx_unit_output_pre_print:
42     \__siunitx_print:nn { number } { \ensuremath {#1} }
43     \__siunitx_unit_output_print:
44   }
45   \group_end:
46 }
47
48 \ExplSyntaxOff

```

File 155 **lwarp-soul.sty**

§ 236 Package **soul**

(Emulates or patches code by MELCHIOR FRANZ.)

pkg soul Emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{soul}[2003/11/17]
 2 \RequirePackage{xcolor}% for \convertcolorspec

Storage for the colors to use:

```

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}{}

```

`\so` `{<text>}`

Basic markup with css:

```
10 \newcommand{\so}[1]{%
11 \LWR@HTMLtextstyle{letter-spacing:.2ex}{letterspacing}{#1}%
12 }
```

`\caps` `{<text>}`

```
13 \newcommand{\caps}[1]{%
14 \LWR@HTMLtextstyle%
15   {font-variant:small-caps;letter-spacing:.1ex}%
16   {capsspacing}{#1}%
17 }
```

`\LWR@soulcolor` `{<text>}{<color>}{<class>}{<colorstyle>}{<FormatWPstyle>}`

Add colors if not empty:

```
18 \newcommand{\LWR@soulcolor}[5]{%
19 \ifcsemt{#2}%
20 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
21 {%
22   \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
23   \LWR@htmlspanclass[#5;#4:\#\LWR@tempcolor]{#3}{#1}%
24 }%
25 }

26 \newcommand{\ul}[1]{%
27 \LWR@soulcolor{#1}{\LWR@soululcolor}{uline}{text-decoration-color}%
28   {text-decoration:underline;text-decoration-skip;}%
29 }
30
31 \newcommand{\st}[1]{
32 \LWR@soulcolor{#1}{\LWR@soulstcolor}{sout}{text-decoration-color}%
33   {text-decoration:line-through}%
34 }
35
36 \newcommand{\hl}[1]{
37 \LWR@soulcolor{#1}{\LWR@soulhlcolor}{highlight}{background-color}%
38   {background:\#F8E800}
39 }
```

Nullified:

```
40 \newcommand*{\soulaccent}[1]{}
41 \newcommand*{\soulregister}[2]{}
42 \newcommand{\sloppyword}[1]{#1}
```

```

43 \newcommand*\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
44 \newcommand*\resetso-{}
45 \newcommand*\capsdef}[5]{}
46 \newcommand*\capsreset-{}
47 \newcommand*\capssave}[1]{}
48 \newcommand*\capssselect}[1]{}
49 \newcommand*\setul}[2]{}
50 \newcommand*\resetul-{}
51 \newcommand*\setuldepth}[1]{}
52 \newcommand*\setuloverlap}[1]{}

```

Set colors:

```

53 \newcommand*\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
54 \newcommand*\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
55 \newcommand*\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}

```

Long versions of the user-level macros:

```

56 \let\textso\so
57 \let\textul\ul
58 \let\texthl\hl
59 \let\textcaps\caps

```

File 156 **lwarp-soulpos.sty**

§ 237 Package **soulpos**

(Emulates or patches code by JAVIER BEZOS.)

Pkg soulpos soulpos is emulated.

for HTML output:

```

1 \RequirePackage{soul}
2 \RequirePackage{soulutf8}
3 \LWR@ProvidesPackageDrop{soulpos}

4 \NewDocumentCommand{\ulposdef}{m o m}{}
5
6 \newdimen\ulwidth
7
8 \newcommand\ifulstarttype[1]{%
9 \expandafter\@secondoftwo%
10 }
11
12 \newcommand\ifulendtype[1]{%
13 \expandafter\@secondoftwo%
14 }

```

```

15
16 \newcommand{\ulstarttype}{0}
17 \newcommand{\ulendtype}{0}
18 \newcommand\ulpostolerance{0}%

```

File 157 **lwarp-soulutf8.sty**

§ 238 Package **soulutf8**

Pkg soulutf8 soulutf8 is emulated.

lwarp's HTML output naturally supports UTF-8 encoding.

for HTML output: 1 \LWR@ProvidesPackageDrop{soulutf8}

File 158 **lwarp-stabular.sty**

§ 239 Package **stabular**

(Emulates or patches code by SIGITAS TOLUŠIS.)

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 159 **lwarp-subfig.sty**

§ 240 Package **subfig**

(Emulates or patches code by STEVEN DOUGLAS COCHRAN.)

Pkg subfig subfig is supported and patched by lwarp.

 **lof/lotdepth** At present, the package options for lofdepth and lotdepth are not working. These counters must be set separately after the package has been loaded.

horizontal spacing In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

for HTML output: Accept all options for lwarp-subfig:

```
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\@addvskip
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 widthOpt heightOpt depthOpt
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 widthOpt heightOpt depthOpt
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@\@capttype}\m@ne
83     \fi
84 \end{minipage}% new
85 \LWR@startpars% new
86 \endgroup\ignorespaces%
87 }%

```

`\sf@subcaption` $\langle 1 \text{ type} \rangle \langle 2 \text{ lof entry} \rangle \langle 3 \text{ caption} \rangle$

```

88 \long\def\sf@subcaption#1#2#3{%
89 \LWR@stoppars% new
90 \ifx \relax#2\relax \else
91     \bgroup
92     \let\label=\@gobble
93     \let\protect=\string
94     \def\@subcaplabel{%
95         \caption@lstfmt{\@nameuse{p#1}}{\@nameuse{the#1}}}%
96     \sf@updatecaptionlist{#1}{#2}{\the\value{\@capttype}}{\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\@capttype name}}%
110 % %                     {\@nameuse{thesub\@capttype}}%
111 % %                     {#3}
112 % % }%
113 % %             \hss
114 % }
115 % }%
116 \@ifundefined{FBsc@max}%
117 %     {\box0}%
118 %     {
119 % \parbox[t]{\the\@tempdima}{%

```

```

120 \LWR@traceinfo{sfsubcap B1}% new
121     \LWR@figcaption% new
122     \caption@make
123         {\@nameuse{sub\@capttype name}}}%
124         {\@nameuse{thesub\@capttype}}}%
125         {#3}
126     \LWR@figcaption% new
127 \LWR@traceinfo{sfsubcap 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` $\{ \langle \textit{caption label} \rangle \} \{ \langle \textit{caption text} \rangle \}$

```

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 }

```

\subfloat@label Patches for \sf@sub@label:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar% new
206 \@ifnextchar(% %) match left parenthesis
207 {\sf@sub@label}
208 {\sf@sub@label(Sub\@capttype\space
209 \ifundefined{thechapter}{\@nameuse{thechapter}\space}%
210 \@nameuse{p@sub\@capttype}%
211 \@nameuse{thesub\@capttype}.)}}

```

Patches for `\subref`.

`\sf@subref` $\langle label \rangle$

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 }
```

`\sf@@subref` $\langle label \rangle$

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.

`\@newsubfloat` [*keys/values*] $\langle float name \rangle$

```
216 \LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat
217
218 \def\@newsubfloat[#1]#2{%
219 \LWR@orig@newsubfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }
```

Pre-defined for figures and tables:

`\l@subfigure` $\langle text \rangle$ $\langle pagenum \rangle$

```
222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
```

`\l@subtable` $\langle text \rangle$ $\langle pagenum \rangle$

```
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
```

File 160 `lwarp-supertabular.sty`

§ 241 Package **supertabular**

(Emulates or patches code by JOHANNES BRAAMS, THEO JURRIENS.)

Pkg `supertabular` `supertabular` is emulated.

for HTML output: `1 \LWR@ProvidesPackageDrop{supertabular}`

△ misplaced alignment
alignment tab character &

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

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

See section 8.7.

```
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 \newcommand*{\LWRST@caption}{}
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 161 **lwarp-syntonly.sty**

§ 242 Package **syntonly**

(Emulates or patches code by FRANK MITTELBACH, RAINER SCHÖPF.)

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*{\syntonly}{}
6
7 \@onlypreamble\syntonly

```

File 162 **lwarp-tables.sty**

§ 243 Package **tables**

(Emulates or patches code by DONALD ARSENEAU.)

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\tablelinesep

```

```
3 \newdimen\arraylinesep
4 \newdimen\extrarulesep
```

File 163 **lwarp-tabularx.sty**

§ 244 Package **tabularx**

(Emulates or patches code by DAVID CARLISLE.)

Pkg tabularx tabularx is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabularx:

```
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 164 **lwarp-tabulary.sty**

§ 245 Package **tabulary**

(Emulates or patches code by DAVID CARLISLE.)

Pkg tabulary tabulary is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabulary.

Column types L, C, R, and J are emulated by lwarp core code.

```
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 165 **lwarp-textarea.sty**

§ 246 Package **textarea**

(Emulates or patches code by ALEXANDER I. ROZHENKO.)

Pkg **textarea** textarea is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textarea}

2 \newcommand\StartFromTextArea{}
3 \newcommand\StartFromHeaderArea{}
4 \newcommand*\RestoreTextArea{}
5 \newcommand*\ExpandTextArea[1][*]{}
6 \let\NCC@restoretextarea\@empty

```

File 166 **lwarp-textcomp.sty**

§ 247 Package **textcomp**

(Emulates or patches code by FRANK MITTELBACH, ROBIN FAIRBAIRNS, WERNER LEMBERG.)

Pkg **textcomp** textcomp is patched for use by lwarp.

§ 247.1 **Limitations**

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

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

§ 247.2 Package loading

for HTML output: `1 \LWR@ProvidesPackagePass{textcomp}`

§ 247.3 Remembering original defintions

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@origtextrquill\textrquill
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@origtextinterrobangdown\textinterrobangdown
16 \let\LWR@origtextpertenthousand\textpertenthousand
17 \let\LWR@origtextbaht\textbaht
18 \let\LWR@origtextdiscount\textdiscount
19 \let\LWR@origtextservicemark\textservicemark
20 \LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
21 \LetLtxMacro\LWR@origcapitalogonek\capitalogonek
22 \LetLtxMacro\LWR@origcapitalgrave\capitalgrave
23 \LetLtxMacro\LWR@origcapitalacute\capitalacute
24 \LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex
25 \LetLtxMacro\LWR@origcapitaltilde\capitaltilde
26 \LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
27 \LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
28 \LetLtxMacro\LWR@origcapitalring\capitalring
29 \LetLtxMacro\LWR@origcapitalcaron\capitalcaron
30 \LetLtxMacro\LWR@origcapitalbreve\capitalbreve
31 \LetLtxMacro\LWR@origcapitalmacron\capitalmacron
32 \LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
33 \LetLtxMacro\LWR@origtextcircled\textcircled

```

§ 247.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for Lua \TeX and Xe \TeX .

§ 247.4.1 pdf \TeX symbols

```

34 \AtBeginDocument{
35 \ifPDFTeX
36 \renewcommand*\textdegree{\HTMLentity{deg}}
37 \renewcommand*\textcelsius{\HTMLunicode{2103}}
38 \renewcommand*\textohm{\HTMLunicode{2126}}
39 \renewcommand*\textmu{\HTMLunicode{00B5}}
40 \renewcommand*\textlquill{\HTMLunicode{2045}}
41 \renewcommand*\textrquill{\HTMLunicode{2046}}
42 \renewcommand*\textcircledP{\HTMLunicode{2117}}
43 \renewcommand*\texttwelveudash{\HTMLunicode{2014}}% emdash
44 \renewcommand*\textthreequartersemdash{\HTMLunicode{2014}}% emdash
45 \renewcommand*\textmho{\HTMLunicode{2127}}
46 \renewcommand*\textnaira{\HTMLunicode{20A6}}
47 \renewcommand*\textpeso{\HTMLunicode{20B1}}
48 \renewcommand*\textrecipe{\HTMLunicode{211E}}
49 \renewcommand*\textinterrobangdown{\HTMLunicode{2E18}}
50 \renewcommand*\textpertenthousand{\HTMLunicode{2031}}
51 \renewcommand*\textbaht{\HTMLunicode{0E3F}}
52 \renewcommand*\textdiscount{\}%
53 \renewcommand*\textservicemark{\HTMLunicode{2120}}
54 \else

```

§ 247.4.2 Xe \TeX and Lua \TeX symbols

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```

55 \renewcommand*\textdegree{\°}
56 \renewcommand*\textcelsius{\C}
57 \renewcommand*\textohm{\Ω}
58 \renewcommand*\textmu{\μ}
59 \renewcommand*\textlquill{\{ }
60 \renewcommand*\textrquill{\} }
61 \renewcommand*\textcircledP{\@}
62 \renewcommand*\texttwelveudash{\--}% emdash
63 \renewcommand*\textthreequartersemdash{\--}% emdash
64 \renewcommand*\textmho{\Ū}
65 \renewcommand*\textnaira{\#}
66 \renewcommand*\textpeso{\P}

```

```

67 \renewcommand*{\textrecipe}{R}
68 \renewcommand*{\textinterrobangdown}{\downarrow}
69 \renewcommand*{\textpertenthousand}{\%..}
70 \renewcommand*{\textbaht}{\฿}
71 \renewcommand*{\textdiscount}{\%}
72 \renewcommand*{\textservicemark}{\textcircled{S}}
73 \fi

```

§ 247.5 HTML diacritics

For HTML, Unicode diacritical marks are used:

```

74 \renewcommand*{\capitalcedilla}[1]{\#1\HTMLUnicode{0327}}
75 \renewcommand*{\capitalogonek}[1]{\#1\HTMLUnicode{0328}}
76 \renewcommand*{\capitalgrave}[1]{\#1\HTMLUnicode{0300}}
77 \renewcommand*{\capitalacute}[1]{\#1\HTMLUnicode{0301}}
78 \renewcommand*{\capitalcircumflex}[1]{\#1\HTMLUnicode{0302}}
79 \renewcommand*{\capitaltilde}[1]{\#1\HTMLUnicode{0303}}
80 \renewcommand*{\capitaldieresis}[1]{\#1\HTMLUnicode{0308}}
81 \renewcommand*{\capitalhungarumlaut}[1]{\#1\HTMLUnicode{30B}}
82 \renewcommand*{\capitalring}[1]{\#1\HTMLUnicode{30A}}
83 \renewcommand*{\capitalcaron}[1]{\#1\HTMLUnicode{30C}}
84 \renewcommand*{\capitalbreve}[1]{\#1\HTMLUnicode{306}}
85 \renewcommand*{\capitalmacron}[1]{\#1\HTMLUnicode{304}}
86 \renewcommand*{\capitaldotaccent}[1]{\#1\HTMLUnicode{307}}

```

`\textcircled` becomes a span with a rounded border:

```

87 \renewcommand*{\textcircled}[1]{%
88 \InlineClass[border: 1px solid \LWR@currenttextcolor]{\textcircled}{\#1}%
89 }
90 }% AtBeginDocument

```

§ 247.6 Inside a lateximage

When a `lateximage` is begun:

```

91 \appto{\LWR@restoreorigformatting}{%
92 \let\textdegree\LWR@origtextdegree%
93 \let\textcelsius\LWR@origtextcelsius%
94 \let\textohm\LWR@origtextohm%
95 \let\textmu\LWR@origtextmu%
96 \let\textlquill\LWR@origtextlquill%
97 \let\textrquill\LWR@origtextrquill%
98 \let\textcircledP\LWR@origtextcircledP%
99 \let\texttwelveudash\LWR@origtexttwelveudash%
100 \let\textthreequartersemdash\LWR@origtextthreequartersemdash%
101 \let\textmho\LWR@origtextmho%
102 \let\textnaira\LWR@origtextnaira%

```

```

103 \let\textpeso\LWR@origtextpeso%
104 \let\textrecipe\LWR@origtextrecipe%
105 \let\textinterrobangdown\LWR@origtextinterrobangdown%
106 \let\textpertenthousand\LWR@origtextpertenthousand%
107 \let\textbaht\LWR@origtextbaht%
108 \let\textdiscount\LWR@origtextdiscount%
109 \let\textservicemark\LWR@origtextservicemark%
110 \LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
111 \LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
112 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
113 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
114 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
115 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
116 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
117 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
118 \LetLtxMacro\capitalring\LWR@origcapitalring%
119 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
120 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
121 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
122 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
123 \LetLtxMacro\textcircled\LWR@origtextcircled%
124 }

```

File 167 **lwarp-textpos.sty**

§ 248 Package **textpos**

(Emulates or patches code by NORMAN GRAY.)

Pkg textpos textpos is emulated.

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 168 **lwarp-theorem.sty**

§ 249 Package **theorem**

(Emulates or patches code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by lwarp.

Table 14: Theorem package — CSS styling of theorems and proofs

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

§ 249.1 **Remembering the theorem style**

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     \theoremstyle{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8     }%
9   {%
10    \theoremstyle{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% new
12  }%
13  \begingroup
14    \csname th@\the\theoremstyle \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 \theorem@style \expandafter
41     \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42     \global \expandafter \let \csname end#1\endcsname \@endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
44   }}
45
46 \gdef\@othm#1[#2]#3{%
47   \expandafter\ifx\csname c@#2\endcsname\relax
48     \@nocounterr{#2}%
49   \else
50     \expandafter\@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 \theorem@style \expandafter
57       \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58       \global \expandafter \let \csname end#1\endcsname \@endtheorem
59       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
60     }%
61   \fi}

```

§ 249.2 CSS patches

The following are patched for css.

These were in individual files thp.sty for plain, thmb.sty for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class theorembody<style>.

Each header is encased in an \InlineClass of class theoremheader.

```

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 169 **lwarp-threeparttable.sty**

§ 250 Package **threeparttable**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg **threeparttable** threeparttable is emulated.

Table notes 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** [*alignment*] To emulate threeparttable:

```
3 \newenvironment*{threeparttable}[1][b]{}{}
```

Env **tablenotes** [*options*]

```
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  <{text}>
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}
```

File 170 **lwarp-tikz.sty**

§ 251 Package **tikz**

(Emulates or patches code by TILL TANTAU.)

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 \begin{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 \end{lateximage}%
```

```

17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
18 {}%
19 {\catcode'\$=\active}%
20 }

21 \end{warpHTML}

```

File 171 **lwarp-titles.sty**

§ 252 Package **titles**

(Emulates or patches code by JAVIER BEZOS.)

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 {<name> [<style>] {<commands>}
2 \NewDocumentCommand{\newpagestyle}{m o m}{}

\renewpagestyle {<name> [<style>] {<commands>}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}

\sethead [<el>] [<ec>] [<er>] {} {<oc>} {<or>}
4 \NewDocumentCommand{\sethead}{o o o m m m}{}

\setfoot [<el>] [<ec>] [<er>] {} {<oc>} {<or>}
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

\settitlemarks *{<names>}
6 \NewDocumentCommand{\settitlemarks}{s m}{}

\headrule
7 \newcommand*{\headrule}{}

<code>\footrule</code>	
	8 <code>\newcommand*\footrule{}</code>
<code>\setheadrule</code>	<code>{\langle length \rangle}</code>
	9 <code>\newcommand*\setheadrule[1]{}</code>
<code>\setfootrule</code>	<code>{\langle length \rangle}</code>
	10 <code>\newcommand*\setfootrule[1]{}</code>
<code>\makeheadrule</code>	
	11 <code>\newcommand*\makeheadrule{}</code>
<code>\makefootrule</code>	
	12 <code>\newcommand*\makefootrule{}</code>
<code>\setmarkboth</code>	<code>{\langle code \rangle}</code>
	13 <code>\newcommand\setmarkboth[1]{}</code>
<code>\widenhead</code>	
	14 <code>\NewDocumentCommand\widenhead{s o o m m}{}</code>
<code>\bottitlemarks</code>	
	15 <code>\newcommand*\bottitlemarks{}</code>
<code>\toptitlemarks</code>	
	16 <code>\newcommand*\toptitlemarks{}</code>
<code>\firsttitlemarks</code>	
	17 <code>\newcommand*\firsttitlemarks{}</code>
<code>\nexttitlemarks</code>	
	18 <code>\newcommand*\nexttoptitlemarks{}</code>
<code>\outertitlemarks</code>	

```

19 \newcommand*\outertitlemarks{}

\innertitlemarks
20 \newcommand*\innertitlemarks{}

\newtitlemark * {<name>}
21 \NewDocumentCommand{\newtitlemark}{s m}{}

\pretitlemark * {<section>} {<text>}
22 \NewDocumentCommand{\pretitlemark}{s m m}{}

\ifsamemark {<group>} {<command>} {<true>} {<false>}
23 \newcommand{\ifsamemark}[4]{}

\setfloathead * [(<.)] [(<.)] [(<.)] {(<.)} {(<.)} {(<.)} {<extra>} [(<which>)]
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}

\setfloatfoot * [(<.)] [(<.)] [(<.)] {(<.)} {(<.)} {(<.)} {<extra>} [(<which>)]
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}

\nextfloathead * [(<.)] [(<.)] [(<.)] {(<.)} {(<.)} {(<.)} {<extra>} [(<which>)]
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}

\nextfloatfoot * [(<.)] [(<.)] [(<.)] {(<.)} {(<.)} {(<.)} {<extra>} [(<which>)]
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}

\newmarkset {<markset>}
28 \newcommand{\newmarkset}[1]{}

\newextramark * {<markset>} {<macro-name>}
29 \NewDocumentCommand{\newextramarkset}{s m m}{}

\botextramarks {<markset>}
30 \newcommand{\botextramarks}[1]{}

```

```

\topextramarks  {\langle markset \rangle}
                 31 \newcommand{\topextramarks}[1]{}

\firstextramarks {\langle markset \rangle}
                 32 \newcommand{\firstextramarks}[1]{}

\nextextramarks  {\langle markset \rangle}
                 33 \newcommand{\nexttopextramarks}[1]{}

\outerextramarks {\langle markset \rangle}
                 34 \newcommand{\outerextramarks}[1]{}

\innerextramarks {\langle markset \rangle}
                 35 \newcommand{\innerextramarks}[1]{}

```

File 172 **lwarp-titleref.sty**

§ 253 Package **titleref**

Pkg titleref titleref is superceded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

File 173 **lwarp-titlesec.sty**

§ 254 Package **titlesec**

(Emulates or patches code by JAVIER BEZOS.)

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

```

\titleref       {\langle label-format \rangle}
                 2 \newcommand*{\titleref}[1]{}

```

`\titleformat*` $\{\langle command \rangle\} \{\langle format \rangle\}$

`\titleformat` $\{\langle command \rangle\} [\langle shape \rangle] \{\langle format \rangle\} \{\langle label \rangle\} \{\langle sep \rangle\} \{\langle before \rangle\} [\langle after \rangle]$

```
3 \newcommand\titleformat{%
4 \ifstar{\ttl@format@s}%
5 {\ttl@format@i}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

```

`\chaptertitlename`

```
8 \@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

```

`\titlespacing` $* \{\langle command \rangle\} \{\langle left \rangle\} \{\langle before \rangle\} \{\langle after \rangle\} [\langle right \rangle]$

```
10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

```

`\filright`

```
11 \newcommand*\filright{}

```

`\filcenter`

```
12 \newcommand*\filcenter{}

```

`\filleft`

```
13 \newcommand*\filleft{}

```

`\fillast`

```
14 \newcommand*\fillast{}

```

`\filinner`

```
15 \newcommand*\filinner{}

```

`\filouter`

```
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}
23 \newcommand\iftitlemeasuring[2] {#2}

\assignpagestyle {command} {pagestyle}
24 \newcommand\assignpagestyle[2] {#2}

\titleclass {name} [startlevel] {class} [cmd]
25 \NewDocumentCommand\titleclass{m o m o}

```

File 174 **lwarp-titletoc.sty**

§ 255 Package **titletoc**

(Emulates or patches code by JAVIER BEZOS.)

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output: 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 o o}{}

```

	5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m o}{}
\contentsmargin	[\langle correction \rangle] {\langle right \rangle}
	6 \newcommand{\contentsmargin}[2] [] {}
\thecontentslabel	
	7 \newcommand*{\thecontentslabel}{thecontentslabel}
\thecontentspage	
	8 \newcommand*{\thecontentspage}{thecontentspage}
\contentslabel	[\langle format \rangle] {\langle space \rangle}
	9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}
\contentspage	[\langle format \rangle]
	10 \newcommand{\contentspage}[1] [] {\thecontentspage}
\contentspush	{\langle text \rangle}
	11 \newcommand{\contentspush}[1] {}
\contentsuse	{\langle name \rangle} {\langle text \rangle}
	12 \newcommand{\contentsuse}[2] {}
\startcontents	[\langle name \rangle]
	13 \newcommand*{\startcontents}[1] [] {}
\stopcontents	[\langle name \rangle]
	14 \newcommand*{\stopcontents}[1] [] {}
\resumecontents	[\langle name \rangle]
	15 \newcommand*{\resumecontents}[1] [] {}
\printcontents	[\langle name \rangle] {\langle prefix \rangle} {\langle start \rangle} {\langle code \rangle}
	16 \newcommand{\printcontents}[4] [] {}

```

\startlist  [<name>] {<list>}
              17 \newcommand{\startlist}[2] [] {}

\stoplist   [<name>] {<list>}
              18 \newcommand{\stoplist}[2] [] {}

\resumelist [<name>] {<list>}
              19 \newcommand{\resumelist}[2] [] {}

\printlist  [<name>] {<list>} {<prefix>} {<code>}
              20 \newcommand{\printlist}[4] [] {}

```

File 175 **lwarp-titling.sty**

§ 256 Package **titling**

(Emulates or patches code by PETER WILSON.)

Pkg titling

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 [55.7](#).

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@origspace{1in}

      \makethanksmark ⇒ \thanksfootmark ⇒ \tmark ⇒
      \@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{\LWR@tagtitle}
53     \@bspretile \@title \@bsposttitle
54     \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
55   }
56   \maketitlehookb
57   {
58     \begin{BlockClass}{author}
59     \renewcommand{\and}{
60       \end{BlockClass}
61       \begin{BlockClass}{oneauthor}
62     }
63     \begin{BlockClass}{oneauthor}
64     \@bspreauthor \@author \@bspostauthor
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 {<series>}
```

Sets the type of footnote marks used by \thanks, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```
84 \renewcommand{\thanksmarkseries}[1]{%
85 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
86 }
```

Set default titlepage thanks footnote marks. See section [55.6](#).

```
87 \if@titlepage
88   \thanksmarkseries{arabic}
89 \else
90   \thanksmarkseries{fnsymbol}
91 \fi
```

File 176 **lwarp-tocbasic.sty**

§ 257 Package **tocbasic**

(Emulates or patches code by MARKUS KOHM.)

Pkg `tocbasic` `tocbasic` is patched for use by `lwarp`.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

for HTML output:

```

1 \LWR@ProvidesPackagePass{tocbasic}

2 \DeclareDocumentCommand{\usetocbasicnumberline}{o}{-}
3 \DeclareDocumentCommand{\DeclareTOCStyleEntry}{o m m}{-}
4 \DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{-}
5 \DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{-}
6 \DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m m}{-}
7 \DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m m}{-}
8 \DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m m}{-}
9 \DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m m}{-}
10 \DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m m}{-}
11 \DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{-}
12 \DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{-}
13 \DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{-}

```

File 177 **lwarp-tocbibind.sty**

§ 258 Package **tocbibind**

(Emulates or patches code by PETER WILSON.)

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`, `anonchp`, `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\@chacntformat##1{%
35         #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
36     }%
37 }
38
39 \renewcommand{\restorechapter}{%
40 \let\@chacntformat\@secntformat%
41 }

```

File 178 **lwarp-tocloft.sty**

§ 259 Package **tocloft**

(Emulates or patches code by PETER WILSON.)

Pkg tocloft tocloft is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

⚠ **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 \TeX 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` `{\langle style \rangle}`
`2 \newcommand{\tocloftpagestyle}[1]{}`

`\cftmarktoc`
`3 \newcommand*\cftmarktoc{}`

`\cfttoctitlefont`
`4 \newcommand*\cfttoctitlefont{}`

`\cftaftertoctitle`
`5 \newcommand*\cftaftertoctitle{}`
`6 \newlength{\cftbeforetoctitleskip}`
`7 \newlength{\cftaftertoctitleskip}`

`\cftmarklof`
`8 \newcommand*\cftmarklof{}`

`\cftloftitlefont`
`9 \newcommand*\cftloftitlefont{}`

`\cftafterloftitle`
`10 \newcommand*\cftafterloftitle{}`
`11 \newlength{\cftbeforeloftitleskip}`
`12 \newlength{\cftafterloftitleskip}`

`\cftmarklot`
`13 \newcommand*\cftmarklot{}`

```
\cftlottitlefont
14 \newcommand*\cftlottitlefont{}

\cftafterlottitle
15 \newcommand*\cftafterlottitle{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

\cftdot
18 \newcommand*\cftdot{.}

\cftdotsep
19 \providecommand*\cftdotsep{1}

\cftnodots
20 \newcommand*\cftnodots{5000}

\cftdotfill {<sep>}
21 \providecommand{\cftdotfill}[1]{}

\cftsetpnumwidth {<length>}
22 \newcommand*\cftsetpnumwidth[1]{}

\cftsetrmarg {<length>}
23 \newcommand*\cftsetrmarg[1]{}

\cftpnumalign {<alignment>}
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{\cftbeforesecskip}
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{\pagenumberon}[1]{ }

\newlistentry [within] {<counter>} {<ext>} {<level-1>}

150 \NewDocumentCommand{\newlistentry}{o m m m}
151 {
152 \IfValueTF{#1}
153 {
154   \newcounter{#2}[#1]
155   \expandafter\edef\csname the#2\endcsname{
156     \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}
157   }
158 }
159 {\newcounter{#2}}
160 \@namedef{l@#2}##1##2{\hypertocfloat{1}{#2}{#3}{##1}{##2}}

```

```

161 \expandafter\newlength\csname cftbefore#2skip\endcsname
162 \expandafter\newlength\csname cft#2indent\endcsname
163 \expandafter\newlength\csname cft#2numwidth\endcsname
164 \@namedef{cft#2font}{}
165 \@namedef{cft#2presnum}{}
166 \@namedef{cft#2aftersnum}{}
167 \@namedef{cft#2aftersnumb}{}
168 \@namedef{cft#2leader}{}
169 \@namedef{cft#2dotsep}{1}
170 \@namedef{cft#2pagefont}{}
171 \@namedef{cft#2afterpnum}{}
172 \@namedef{toclevel@#2}{#4}
173 \@namedef{cft#2fillnum}##1{}
174 }

```

`\newlistof` [*within*] {*type*} {*ext*} {*listofname*}

Emulated through the `\newfloat` mechanism.

```

175 \NewDocumentCommand{\newlistof}{o m m}
176 {%
177 \IfValueTF{#1}
178 {\newlistentry[#1]{#2}{#3}{0}}
179 {\newlistentry{#2}{#3}{0}}
180 \@namedef{ext@#2}{#3}
181 \newcounter{#3depth}
182 \setcounter{#3depth}{1}
183 \@namedef{cftmark#3}{}
184 \@namedef{listof#2}{\listof{#2}{#4}}
185 \@namedef{@cftmake#3title}{}
186 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
187 \expandafter\newlength\csname cftafter#3titleskip\endcsname
188 \@namedef{cft#3titlefont}{}
189 \@namedef{cftafter#3title}{}
190 \@namedef{cft#3prehook}{}
191 \@namedef{cft#3posthook}{}
192 }

```

`\cftchapterprecis` {*text*}

```

193 \newcommand{\cftchapterprecis}[1]{%
194 \cftchapterprecishere{#1}
195 \cftchapterprecistoc{#1}}
196 \newcommand{\cftchapterprecishere}[1]{%
197 \begin{quote}\textit{#1}\end{quote}}
198 \newcommand{\cftchapterprecistoc}[1]{
199 \addtocontents{toc}{%
200 {
201 \protect\begin{quote}#1\protect\end{quote}}
202 }

```

203 }

File 179 **lwarp-tocstyle.sty**§ 260 Package **tocstyle**

Pkg tocstyle tocstyle is ignored.

 **Not yet tested!** [Please send bug reports!](#)

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tocstyle}

2 \newcommand*\usetocstyle}[2] [] {}
3 \newcommand*\deactivatetocstyle}[1] [] {}
4 \newcommand*\reactivatetocstyle}[1] [] {}
5 \NewDocumentCommand{\settocfeature}{o o m m}{}
6 \NewDocumentCommand{\settocstylefeature}{o m m}{}
7 \NewDocumentCommand{\newtocstyle}{o o m m}{}
8 \newcommand*\aliastoc}[2] {}
9 \newcommand*\showtoc}[2] [] {}
10 \newcommand{\iftochasdepth}[4] {}

```

File 180 **lwarp-todo.sty**§ 261 Package **todo***(Emulates or patches code by FEDERICO GARCIA.)*

Pkg todo todo is patched for use by lwarp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{todo}

2 \renewcommand\todoitem[2] {%
3   \refstepcounter{todo}%
4   \item[%
5     \HTMLunicode{2610} \quad
6     \ref{todopage:\thetodo}
7   ] : {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
8   \label{todobl:\thetodo}%
9 }%
10
11 \renewcommand\doneitem[2] {%
12   \stepcounter{todo}%
13   \item[%

```

```

14      \HTMLUnicode{2611} \quad
15      \ref{todopage:\thetodo}
16    ] \@nameuse{@done\the\c@todo}:
17      {\todoformat\ifx#1\todomark\else\textbf{#1} \fi}#2%
18 }
19
20 \xpatchcmd{\@displaytodo}
21   {\todoformat #1}{\todoformat \textbf{#1}}{}
22   {\PackageWarning{lwarp-todo}{Unable to patch @displaytodo.}}
23
24 \xpatchcmd{\@displayfulltodo}
25   {\todoformat #1}{\todoformat \textbf{#1}}{}
26   {\PackageWarning{lwarp-todo}{Unable to patch @displayfulltodo.}}
27
28 \patchcmd{\todoenv}{\itshape see text.}{\textit{see text.}}{}
29   {\PackageWarning{lwarp-todo}{Unable to patch todoenv.}}
30
31 \patchcmd{\astodos}{\todoformat #1}{\todoformat \textbf{#1}}{}
32   {\PackageWarning{lwarp-todo}{Unable to patch astodos.}}
33
34 \AtBeginDocument{
35   \crefname{todo}{todo}{todos}
36   \Crefname{todo}{Todo}{Todos}
37 }

```

File 181 **lwarp-todonotes.sty**

§ 262 Package **todonotes**

(Emulates or patches code by HENRIK SKOV MIDTIBY.)

Pkg **todonotes** todonotes is emulated.

The documentation for todonotes and luatodonotes have an example with a todo inside a caption. If this example does not work it will be necessary to move the todo outside of the caption.

for HTML output:

```

1 \LWR@ProvidesPackagePass{todonotes}

2 \if@todonotes@disabled
3 \else
4
5 \newcommand{\ext@todo}{tdo}
6
7 \renewcommand{\l@todo}[2]{\hypertocfloat{1}{\todo}{ldo}{#1}{#2}}
8
9

```

```

10 \renewcommand{\@todonotes@drawMarginNoteWithLine}{
11 \fcolorbox
12   {\@todonotes@currentbordercolor}
13   {\@todonotes@currentbackgroundcolor}
14   {\arabic{\@todonotes@numberoftodonotes}}
15 \marginpar{\@todonotes@drawMarginNote}
16 }
17
18 \renewcommand{\@todonotes@drawInlineNote}{%
19 \fcolorboxBlock%
20   {\@todonotes@currentbordercolor}%
21   {\@todonotes@currentbackgroundcolor}%
22   {%
23     \if@todonotes@authorgiven%
24     {\@todonotes@author:\,%
25     \fi%
26     \@todonotes@text%
27   }%
28 }
29
30 \renewcommand{\@todonotes@drawMarginNote}{%
31   \if@todonotes@authorgiven%
32     \@todonotes@author\par%
33   \fi%
34   \arabic{\@todonotes@numberoftodonotes}: %
35   \fcolorbox%
36   {\@todonotes@currentbordercolor}%
37   {\@todonotes@currentbackgroundcolor}%
38   {%
39     \@todonotes@sizecommand%
40     \@todonotes@text %
41   }%
42 }%
43
44 \renewcommand{\@todonotes@drawLineToRightMargin}{}
45
46 \renewcommand{\@todonotes@drawLineToLeftMargin}{}
47
48 \renewcommand{\missingfigure}[2] []{%
49 \setkeys{todonotes}{#1}%
50 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
51 \fcolorboxBlock%
52   {\@todonotes@currentbordercolor}%
53   {\@todonotes@currentfigcolor}%
54   {%
55     \setlength{\fboxrule}{4pt}%
56     \fcolorbox{red}{white}{Missing figure} \quad #2%
57   }
58 }
59

```

```

60 \LetLtxMacro\LWRTODONOTES@orig@todo\@todo
61
62 \RenewDocumentCommand{\@todo}{o m}{%
63 \begingroup%
64 \renewcommand*\phantomsection}{}%
65 \IfValueTF{#1}{%
66   \LWRTODONOTES@orig@todo[#1]{#2}%
67 }{%
68   \LWRTODONOTES@orig@todo{#2}%
69 }
70 \endgroup%
71 }
72
73 \fi% \if@todonotes@disabled

```

File 182 **lwarp-transparent.sty**

§ 263 Package **transparent**

(Emulates or patches 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 183 **lwarp-trivfloat.sty**

§ 264 Package **trivfloat**

(Emulates or patches code by JOSEPH WRIGHT.)

Pkg `trivfloat` `trivfloat` is forced to use the built-in `lwarp` emulation for floats.

Discard all options for `lwarp-trivfloat`. This tells `trivfloat` not to use `floatrow` or `memoir`.

To create a new float type and change its name:

```
\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}
```

```
1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}
```

`\tfl@chapter@fix` Nullified at the beginning of the document. Is used by `trivfloat` to correct float chapter numbers, but is not needed for `lwarp`.

for HTML output: `3 \begin{warpHTML}`

```
4 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}
```

```
5 \end{warpHTML}
```

§ 264.1 **Combining `\newfloat`, `\trivfloat`, and `algorithmicx`**

for HTML & PRINT: `6 \begin{warpall}`

For both print and HTML output:

- ⚠ When using `float`, `trivfloat`, or `algorithmicx` at the same time, be aware of conflicting file usage. `algorithmicx` uses `.loa`. `trivfloat` by default starts with `.loa` and goes up for additional floats, skipping `.lof` and `.lot`.
- ⚠ When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid `.loa` used by `algorithmicx`, and any files used by `trivfloat`. Also avoid using `.lof` and `.lot`.
- ⚠ When using `\trivfloat`, you may force it to avoid conflicting with `algorithmicx` by starting `trivfloat`'s file extensions with `.lob`:

```
\makeatletter
\setcounter{\tfl@float@cnt}{1} % start trivfloats with .lob
\makeatletter
```

```
7 \end{warpall}
```

File 184 **lwarp-typearea.sty**

§ 265 Package **typearea**

(Emulates or patches code by MARKUS KOHM.)

Pkg typearea typearea is emulated.

This package may be loaded standalone, but is also loaded automatically if koma-script classes are in use. `\DeclareDocumentCommand` is used to overwrite the koma-script definitions.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{typearea}

2 \DeclareDocumentCommand{\typearea}{o m}{}
3 \DeclareDocumentCommand{\recalctypearea}{}{}
4 \@ifundefined{footheight}{\newlength\footheight}{}
5 \DeclareDocumentCommand{\areaset}{o m m}{}
6 \DeclareDocumentCommand{\activateareas}{}{}
7 \DeclareDocumentCommand{\storeareas}{m}{}
8 \DeclareDocumentCommand{\BeforeRestoreareas}{s m}{}
9 \DeclareDocumentCommand{\AfterRestoreareas}{s m}{}
10 \DeclareDocumentCommand{\AfterCalculatingTypearea}{s m}{}
11 \DeclareDocumentCommand{\AfterSettingArea}{s m}{}

```

File 185 **lwarp-ulem.sty**

§ 266 Package **ulem**

(Emulates or patches 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 \LWR@HTMLtextstyle%

```

```
5 {text-decoration:underline;text-decoration-skip}%
6 {uline}#{1}%
7 }
8
9 \NewDocumentCommand{\uuline}{+m}{%
10 \LWR@HTMLtextstyle%
11 {%
12     text-decoration:underline;text-decoration-skip;%
13     text-decoration-style:double%
14 }%
15 {uuline}#{1}%
16 }
17
18 \NewDocumentCommand{\uwave}{+m}{%
19 \LWR@HTMLtextstyle%
20 {%
21     text-decoration:underline;text-decoration-skip;%
22     text-decoration-style:wavy%
23 }%
24 {uwave}#{1}%
25 }
26
27 \NewDocumentCommand{\sout}{+m}{%
28 \LWR@HTMLtextstyle%
29 {text-decoration:line-through}%
30 {sout}#{1}%
31 }
32
33 \NewDocumentCommand{\xout}{+m}{%
34 \LWR@HTMLtextstyle%
35 {text-decoration:line-through}%
36 {xout}#{1}%
37 }
38
39 \NewDocumentCommand{\dashuline}{+m}{%
40 \LWR@HTMLtextstyle%
41 {%
42     text-decoration:underline;%
43     text-decoration-skip;%
44     text-decoration-style:dashed%
45 }%
46 {dashuline}#{1}%
47 }
48
49 \NewDocumentCommand{\dotuline}{+m}{%
50 \LWR@HTMLtextstyle%
51 {%
52     text-decoration:underline;%
53     text-decoration-skip;%
54     text-decoration-style:dotted%
```

```

55   }%
56   {dotuline}{#1}%
57 }

```

Nullified parameters:

```

58 \NewDocumentCommand{\ULthickness}{-}{-}
59 \newlength{\ULdepth}

```

Nullified/emulated macros:

```

60 \NewDocumentCommand{\markoverwith}{m}{-}
61 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}

```

`\useunder` only works with `\textbf`, etc, but not `\bfseries`, etc.

```

62 \NewDocumentCommand{\useunder}{m m m}{%
63 \relax%
64 \ifx\relax#3\relax\else % argumentative command
65   \def#3{#1}\MakeRobust{#3}\fi
66 }

```

Triggered by package options, also available for the users:

```

67 \newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
68 \newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
69 \ULforem% default

```

Package options:

```

70 \DeclareOption{normalem}{\normalem}
71 \DeclareOption{ULforem}{\ULforem}
72 \DeclareOption{normalbf}{-}
73 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}

```

Emulate the original package:

```

74 \LWR@ProvidesPackageDrop{ulem}

```

File 186 `lwarp-upref.sty`

§ 267 Package **upref**

Pkg upref Ignored.

for HTML output:

Discard all options for `lwarp-upref`:

```
1 \LWR@ProvidesPackageDrop{upref}
```

File 187 **lwarp-verse.sty**

§ 268 Package **verse**

(Emulates or patches code by PETER WILSON.)

Pkg `verse` `verse` is supported and patched by `lwarp`.

for HTML output: Pass all options for `lwarp-verse`:

```
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.

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 {%

```

Use the original list environment inside a <pre> to attempt to preserve formatting.

```

6 \LWR@restoreoriglists%

```

Pkg verse The verse or memoir packages can place stanza numbers to the left with their
Pkg memoir \flagverse command. Do not allow them to go into the left margin, which would
\flagverse cause pdfcrop to crop the entire page further to the left:

```

Len \leftskip
7 \ifdef{\vleftskip}{%
8 \setlength{\vleftskip}{\HTMLvleftskip}
9 \setlength{\leftmargini}{\HTMLleftmargini}
10 }{}
11 \LWR@forcenewpage
12 \LWR@atbeginverbatim{verse}
13 \unskip\LWR@origvspace{-\baselineskip}
14 }

```

After the end of the verse environment, which places the <pre> tag at the regular left margin:

```

15 \AtEndEnvironment{verse}{
16 \LWR@afterendverbatim
17 }

```

Patch to place poemtitle inside an HTML 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 \LWR@traceinfo{Finished patching verse.}
27 }% AfterEndPreamble

```

File 188 **lwarp-wallpaper.sty**

§ 269 Package **wallpaper**

(Emulates or patches code by MICHAEL H.F. WILKINSON.)

Pkg wallpaper wallpaper is emulated.

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[1]{}
17 \newlength\wpXoffset
18 \newlength\wpYoffset
```

File 189 **lwarp-watermark.sty**

§ 270 Package **watermark**

(Emulates or patches code by ALEXANDER I. ROZHENKO.)

Pkg watermark watermark is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{watermark}

2 \newcommand\watermark[1]{}
3 \newcommand\leftwatermark[1]{}
4 \newcommand\rightwatermark[1]{}
5 \newcommand\thiswatermark[1]{}
6 \newcommand\thispageheading[1]{}

```

File 190 **lwarp-wrapfig.sty**

§ 271 Package **wrapfig**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg wrapfig wrapfig is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{wrapfig}

2 \newcommand*\LWR@wrapposition}{
3
4 \newcommand*\LWR@subwrapfigure}[2]{%
5 \renewcommand*\LWR@wrapposition}{}%
6 \ifthenelse{%
7   \equal{#1}{r}\OR\equal{#1}{R}\OR%
8   \equal{#1}{o}\OR\equal{#1}{O}%
9 }%
10 {\renewcommand*\LWR@wrapposition}{float:right}}%
11 {\renewcommand*\LWR@wrapposition}{float:left}}%
12 \setlength{\LWR@templengthone}{#2}%
13 \uselengthunit{PT}%
14 \LWR@BlockClassWP{%
15   width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition; %
16   margin:10pt%
17 }%
18 {%
19   width:\rndprintlength{\LWR@templengthone}; \LWR@wrapposition%
20 }%
21 {marginblock}%
22 }
23
24
25 \NewDocumentEnvironment{wrapfigure}{o m o m}
26 {%
27 \LWR@subwrapfigure{#2}{#4}%
28 \captionsetup{type=figure}%
29 }
30 {%
31 \endLWR@BlockClassWP%
32 }
33
34
35 \NewDocumentEnvironment{wraptable}{o m o m}
36 {%

```

```

37 \LWR@subwrapfigure{#2}{#4}%
38 \captionsetup{type=table}%
39 }
40 {%
41 \endLWR@BlockClassWP%
42 }
43
44
45 \NewDocumentEnvironment{wrapfloat}{m o m o m}
46 {%
47 \LWR@subwrapfigure{#3}{#5}%
48 \captionsetup{type=#1}%
49 }
50 {%
51 \endLWR@BlockClassWP%
52 }
53
54 \newlength{\wrapoverhang}

```

File 191 **lwarp-xcolor.sty**

§ 272 Package **xcolor**

(Emulates or patches code by DR. UWE KERN.)

Pkg xcolor xcolor is supported by lwarp.

§ 272.1 **Limitations**

\colorboxBlock and \fcolorboxBlock \colorboxBlock and \fcolorboxBlock are provided for increased HTML compatibility, and they are identical to \colorbox and \fcolorbox in print mode. In HTML mode they place their contents into a <div> instead of a . 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.

§ 272.2 **Xcolor definitions: location and timing**

The `lwarp` core and its `lwarp-xcolor` package are tightly integrated to allow comparable results for `print`, `HTML` and `print` inside an `HTML lateximage`. This requires a number of definitions and 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 73. 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 365.

`\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 647.

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 647.

HTML lateximage: Remembers and reuses the print version.

`\colorboxBlock:`

Print: Becomes `\colorbox`.

HTML: Newly defined by lwarp-xcolor to use a `<div>`, page 648.

HTML lateximage: Remembers and reuses the print version `\colorbox`.

`\fcolorbox:`

Print: Modified to allow a background of none.

`\LWRprint@fcolorbox` at section 73

HTML: Redefined by lwarp-xcolor, page 648.

HTML lateximage: Remembers and reuses the print version.

`\fcolorboxBlock:`

Print: Becomes `\fcolorbox`. Section 73

HTML: Newly defined by lwarp-xcolor to use a `<div>`, page 649.

HTML lateximage: Remembers and reuses the print version `\fcolorbox`.

`fcolorminipage:`

Print: Newly defined in the lwarp core.

`LWRprint@fcolorminipage` at section 73

HTML: Newly defined by lwarp-xcolor, page 649.

HTML lateximage: Uses the print version.

`\boxframe:`

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 650.

HTML lateximage: Remembers and reuses the print version.

§ 272.3 Package loading

for HTML output:

```
1 \LWR@ProvidesPackagePass{xcolor}
2 \begin{warpHTML}
```

§ 272.4 Remembering and restoring original definitions

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 }
```

§ 272.5 HTML color style

`\LWR@tempcolor` The color converted to HTML colorspace.

```
19 \newcommand*{\LWR@tempcolor}{}
20 \newcommand*{\LWR@tempcolortwo}{}

```

Sets `\LWR@tempcolor` to the current color.

```
\LWR@findcurrenttextcolor
21 \newcommand*{\LWR@findcurrenttextcolor}{%
22 \protect\colorlet{\LWR@current@color}{.}%
23 \protect\convertcolourspec{named}{\LWR@current@color}{HTML}\LWR@tempcolor%
24 }
```

Prints a color style for the current color.

```
\LWR@currenttextcolorstyle
25 \newcommand*{\LWR@currenttextcolorstyle}{%
26 \LWR@findcurrenttextcolor%
```

```

27 \ifdefstring{\LWR@tempcolor}{000000}%
28 {}%
29 {color: \#\LWR@tempcolor ; }%
30 }

```

`\LWR@textcurrentcolor` `{\langle text \rangle}` Like `\textcolor` but uses the current `\color` instead.

```

31 \newcommand*\LWR@textcurrentcolor[1]{%
32 \begingroup%
33 \LWR@FBcancel%
34 \LWR@findcurrenttextcolor%
35 \InlineClass[color:\#\LWR@tempcolor]{textcolor}{%
36   \renewcommand*\LWR@currenttextcolor{\#\LWR@tempcolor}%
37   #1%
38 }%
39 \endgroup%
40 }

```

`\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.

```

41 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
42 \begingroup%
43 \LWR@FBcancel%

```

Use the `xcolor` package to convert to an HTML color space:

```

44 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%

```

Print the converted color:

```

45 \#\LWR@tempcolor%
46 \endgroup%
47 }

```

`\LWR@backgroundcolor` `[\langle model \rangle]{\langle color \rangle}{\langle text \rangle}`

Similar to `\textcolor`, but prints black text against a color background.

Converted into an HTML hex color span.

```

48 \NewDocumentCommand{\LWR@backgroundcolor}{O{named} m m}{%
49 \begingroup%
50 \LWR@FBcancel%
51 \InlineClass[background:\LWR@colorstyle{#1}{#2}]{backgroundcolor}{%
52 #3%
53 }%
54 \endgroup%
55 }

```

§ 272.6 HTML border

`\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.

```
56 \newcommand*\LWR@borderpadding[2]{%
57 \uselengthunit{PT}%
58 border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
59 padding:\rndprintlength{\fboxsep}%
60 }
```

§ 272.7 High-level macros

`\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.

```
61 \RenewDocumentCommand{\textcolor}{O{named} m m}{%
62 \begingroup%
63 \LWR@FBcancel%
64 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
65 \renewcommand*\LWR@currenttextcolor}{#\LWR@tempcolor}%
66 #3%
67 }%
68 \endgroup%
69 }
```

`\pagecolor` $[\langle model \rangle]$ $\langle color \rangle$

Ignored. Use `\CSSFilename` instead.

```
70 \renewcommand*\pagecolor[2][named]{}
```

`\nopagecolor` Ignored.

```
71 \renewcommand*\nopagecolor{}
```

`\colorbox` $[\langle model \rangle]$ $\langle color \rangle$ $\langle text \rangle$

Converted into an HTML hex background color ``.

```

72 \RenewDocumentCommand{\colorbox}{0{named} m +m}{%
73 \begingroup%
74 \LWR@FBcancel%
75 \uselengthunit{PT}%
76 \InlineClass[%
77 background:\LWR@colorstyle{#1}{#2} ; %
78 padding:\rndprintlength{\fboxsep}%
79 ]{colorbox}{#3}%
80 \endgroup%
81 }

```

`\colorboxBlock` [*model*] {*color*} {*text*}

Converted into an HTML hex background color <div>.

```

82 \NewDocumentCommand{\colorboxBlock}{0{named} m +m}{%
83 \begingroup%
84 \LWR@FBcancel%
85 \uselengthunit{PT}%
86 \begin{BlockClass}%
87 background:\LWR@colorstyle{#1}{#2} ; %
88 padding:\rndprintlength{\fboxsep}%
89 ]{colorboxBlock}
90 #3
91 \end{BlockClass}%
92 \endgroup%
93 }

```

`\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.

```

94 \RenewDocumentCommand{\fcolorbox}{0{named} m 0{named} m +m}{%
95 \LWR@traceinfo{HTML fcolorbox #2 #4}%
96 \begingroup%
97 \LWR@FBcancel%
98 \uselengthunit{PT}%
99 \LWR@forceminwidth{\fboxrule}%
100 \ifthenelse{\equal{#4}{none}}%
101 {% no background color
102   \InlineClass[%
103     \LWR@borderpadding{#1}{#2}%
104   ]{fcolorbox}{#5}%
105 }%
106 {% yes background color

```

```

107 \InlineClass[%
108 \LWR@borderpadding{#1}{#2} ; %
109 background:\LWR@colorstyle{#3}{#4}%
110 ]{fcolorbox}{#5}%
111 }%
112 \endgroup%
113 }

```

`\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.

```

114 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{named} m +m}{%
115 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
116 \begingroup%
117 \LWR@FBcancel%
118 \uselengthunit{PT}%
119 \LWR@forceminwidth{fboxrule}%
120 \ifthenelse{\equal{#4}{none}}{%
121 {% no background color
122 \begin{BlockClass}[%
123 \LWR@borderpadding{#1}{#2}%
124 ]{fcolorboxBlock}
125 #5
126 \end{BlockClass}%
127 }%
128 {% yes background color
129 \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
130 \begin{BlockClass}[%
131 background:\#\LWR@tempcolortwo\ ; %
132 \LWR@borderpadding{#1}{#2}%
133 ]{fcolorboxBlock}
134 #5
135 \end{BlockClass}%
136 }%
137 \endgroup%
138 \LWR@traceinfo{HTML fcolorboxBlock done}%
139 }

```

Creates a framed HTML `<div>` around its contents.

A print-output version is defined in the lwarp core: [section 73](#)

`\LWR@subfcolorminipage` *{framemodel}* *{framecolor}* *{background tag}* *{height}*

```

140 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
141 \begin{BlockClass}[%
142 #3%
143 \LWR@borderpadding{#1}{#2} ; %
144 \IfValueT{#4}{height:\rndprintlength{\LWR@tempheight} ; }%
145 width:\rndprintlength{\LWR@tempwidth}%
146 ]{fcolorminipage}%
147 }

```

Env fcolorminipage [*1:framemodel*] [*2:framecolor*] [*3:boxmodel*] [*4:boxcolor*] [*5:align*] [*6:height*]
 [*7:inner-align*] [*8:width*]

```

148 \NewDocumentEnvironment{fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
149 {%
150 \LWR@FBcancel%
151 \setlength{\LWR@tempwidth}{#8}%
152 \IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
153 \uselengthunit{PT}%
154 \LWR@forceminwidth{\fboxrule}%
155 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
156 \ifthenelse{\equal{#4}{none}}%
157 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
158 {%
159   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
160   \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ; }{#6}%
161 }%
162 }
163 {\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.

```

164 \renewcommand*{\boxframe}[3]{%
165 {%
166 \setlength{\LWR@tempwidth}{#1}%
167 \setlength{\LWR@tempheight}{#2}%
168 \addtolength{\LWR@tempheight}{#3}%
169 \uselengthunit{PT}%
170 \LWR@forceminwidth{\fboxrule}%
171 \InlineClass[%
172 display:inline-block ; %
173 border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{ } ; %
174 width:\rndprintlength{\LWR@tempwidth} ; %
175 height:\rndprintlength{\LWR@tempheight}%
176 ]{boxframe}{}%

```

```

177 }%
178 }

179 \end{warpHTML}

```

File 192 **lwarp-xfrac.sty**

§ 273 Package **xfrac**

(Emulates or patches code by THE L^AT_EX3 PROJECT.)

Pkg xfrac Supported by adding xfrac instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

 **font size** In the user's document preamble, lwarp should be loaded after font-related setup. During HTML conversion, this font is used by lwarp to generate its initial PDF output containing HTML tags, later to be converted by pdftotext to a plain text file. While the text may be in any font which pdftotext can read, the math is directly converted into SVG images using this same user-selected font. xfrac below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine \xfracHTMLfontsize with a different em size.

\sfrac [*instance*] {*num*} [*sep*] {*denom*}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. \scalebox is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a lateximage, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

\xfracHTMLfontsize User-redefinable macro which controls the font size of the fraction.

```
3 \newcommand*\xfracHTMLfontsize{.6em}
```

```
4 \end{warpall}
```

for HTML output: 5 \begin{warpHTML}

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*\LWR@htmlsmallfontstart}{%
```

```

7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"{}}%
8 \LWR@nestspan%
9 %
10 }
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 193 **lwarp-xltxtra.sty**

§ 274 Package **xltxtra**

(Emulates or patches code by WILL ROBERTSON, JONATHAN KEW.)

Pkg xltxtra xltxtra is emulated.

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 194 **lwarp-xmpincl.sty**

§ 275 Package **xmpincl**

(Emulates or patches code by MAARTEN SNEEP.)

Pkg xmpincl Emulated.

for HTML output: Discard all options for lwarp-xmpincl:

```

1 \LWR@ProvidesPackageDrop{xmpincl}
2 \newcommand*{\includemp}[1]{}

```

File 195 **lwarp-xtab.sty**

§ 276 Package **xtab**

(Emulates or patches code by PETER WILSON.)

Pkg xtab xtab is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{xtab}

 **misplaced alignment tab character &** For `\tablefirsthead`, etc., enclose them as follows:

```

\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars

```

See section 8.7.

```

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 \newcommand*{\LWRXT@caption}{%
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 196 **lwarp-zwpagelayout.sty**

§ 277 Package **zwpagelayout**

(Emulates or patches code by ZDENĚK WAGNER.)

Pkg zwpagelayout zwpagelayout is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{zwpagelayout}

2 \def\noBboxes{}
3 \@onlypreamble\noBboxes
4
5 \expandafter\ifx\csname definecolor\endcsname\relax \else
6 \definecolor{cmykblack}{cmyk}{0,0,0,1}
7 \definecolor{grblack}{gray}{0}
8% \ifzwpl@redefineblack
9% \definecolor{black}{cmyk}{0,0,0,1}\color{black}
10% \fi
11 \definecolor{cmykred}{cmyk}{0,1,1,0}
12 \definecolor{cmykgreen}{cmyk}{1,0,1,0}
13 \definecolor{cmykblue}{cmyk}{1,1,0,0}
14 \definecolor{rgbred}{rgb}{1,0,0}
15 \definecolor{rgbgreen}{rgb}{0,1,0}
16 \definecolor{rgbblue}{rgb}{0,0,1}
17% \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\Z@evenwarning}
41 \def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning
42
43 \def\clap#1{#1}
44
45 \def\CropFlap{2in}
46 \def\CropSpine{1in}
47 \def\CropXSpine{1in}
48 \def\CropXtrim{.25in}
49 \def\CropYtrim{.25in}
50 \def\UserWidth{5in}
51 \def\UserLeftMargin{1in}
52 \def\UserRightMargin{1in}
53 \def\UserTopMargin{1in}
54 \def\UserBotMargin{1in}
55 \def\thePageNumber{\#\,\arabic{page}}
56 \ifXeTeX
57 \def\ifcaseZWdriver{\ifcase2}
58 \else
59 \def\ifcaseZWdriver{\ifcase1}
60 \fi
61 \DeclareRobustCommand\ZWifdriver[2]{}

```

Change History and Index

§ 277 Change History

v0.10	General: 2016/03/08 Initial version . . . 1	Docs: Table: Float data structures. 349
v0.11	General: 2016/03/11 1	Docs: Trademarks section. 432
	Added section: Operating-System	Docs: Troubleshooting
	portability. 130	cross-references. 121
	Added section: Selecting the	Test Suite: Assigned cleveref name
	operating system. 80	for Test Float. 1
	Test Suite: limages and index in	Test Suite: Floatrow 1
	README.txt 1	
	Test Suite: MS-Windows in	
	README.txt 1	
v0.12	<code>\LWR@newhtmlfile</code> : Bugfix: TOC	
	with numbered files. 249	
	General: 2016/03/14 1	
	Global: Uses <code>\p@(type)</code> in float	
	captions. 1	
	Test Suite: Sub-figures 1	
v0.13	<code>\CaptionSeparator</code> : Fix for newer	
	babel package. 352	
	<code>\LWR@LwarpStart</code> : <code>\up</code> and <code>\fup</code> . . 265	
	General: 2016/03/24 1	
	Removed package: subfig 1	
	Test Suite: Ordinals, Subcaption . . 1	
	<code>tikzpicture</code> : Fix dollar-redefined	
	bug for newer package. 608	
v0.14	<code>\LWR@htmlsectionfilename</code> : Fix:	
	Links to home page. 217	
	General: 2016/03/31 1	
	floatrow: Added. 479	
	Docs: Commands for a successful	
	HTML conversion. 85	
	Docs: Commands into a warpprint	
	environment. 82	
	Docs: Newclude limitations. . . . 108	
	Docs: Table: Cross-referencing	
	data structures. 339	
		General: 2016/04/06 1
		Added 484
		Ampersand (&): Fixed handling
		when passed as an argument. . . 296
		Docs: Added warning icons for
		items needing special attention. 126
		Docs: Clarify print/HTML output. 81
		Docs: Moved the supported
		functions table to the
		introduction. 43
		Files: <code>lwarp_formal.css</code> added. . . . 1
		Fix: steps counter 484
		Fixed & handling. 482
		Test Suite: <code>test_suite_formal.css</code>
		file added. 1
		v0.15
		General: 2016/04/11 1
		<code>\titlingpage</code> : Improved
		print-output spacing. 273
		Added XeLaTeX, LuaLaTeX
		support. 140
		Docs: Font and UTF-8 support. . . 70
		Docs: Moved location of
		<code>\usepackage{lwarp}</code> 72
		Docs: Text not converting. 121
		Fix: <code>amsmath</code> options clash 144
		Fix: <code>newtxmath</code> compatibility. . . 144
		Lwarp no longer selects fonts. 70, 140
		Removed package: suffix 1
		Test Suite: Improved titlingpage. 273
		Test Suite: Lwarp no longer selects
		fonts. 1

Test Suite: Supports XeLaTeX, LuaLaTeX.	1	verse: Supports verse, memoir packages.	638
xfrac: Adjusted for the use of any font.	652	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage.	406
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page.	217	\HTMLFilename: Docs: Escape filename underscores.	216
General: 2016/04/14	1	\HomeHTMLFilename: Docs: Escape filename underscores.	216
mdframed: Added.	529	\LWR@LwarpStart: Enabled \\ equal to \newline.	264
Test Suite: Fix: Print-version front-matter page numbers.	1	\LWR@doubledollar: MathJax support.	371
Test Suite: Mdframed	1	\LWR@filestart: lwarp_mathjax.txt loaded.	262
v0.18		\LWR@hspace: Fix: \hspace length computations.	423
\LWR@hspace: \hspace supported.	423	\LWR@minipagestartpars: Surpresses paragraph tags between minipages.	422
\LWR@includegraphicsb: Add: svgz file extension.	499	\LWR@singledollar: MathJax support.	371
em, ex, %, px dimensions preserved.	499	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size.	387
Fix: \linewidth, \textwidth, \textheight inside a minipage.	499	\minipagefullwidth: Added: No width tag for the next minipage in HTML.	406
Improved HTML output linebreaks.	499	\rule: Added	428
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic.	356	\warpHTMLonly: Added.	134
\LWR@newhtmlfile: sideroc after title, improving responsive design.	248	\warpprintonly: Replaces \rowprintedonly.	134
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic.	267	\xfracHTMLfontsize: Added.	651
\LWR@subhyperref: Improved HTML output linebreaks.	347	General: 2016/06/08	1
\LWR@subhyperrefclass: Improved HTML output linebreaks.	347	Added optional args.	543
\LWR@subinlineimage: Surpress extra space.	348	Avoids MathJax.	370
General: 2016/05/19	1	cleveref: Loaded \AtEndPreamble.	402
File: lwarp.css: Improved TOC outline display.	1	CSS for table note item.	607
Files: lwarp.css and lwarp_formal.css: Improved responsive design.	1	Docs: Math options.	72
Microtype disabled during HTML generation	141	Docs: Table: Cross-referencing data structures, updated.	339
PDF Unicode input characters.	127	File: lwarp_mathjax.txt added.	1
Test Suite: Verse package	1	File: lwarp.css: tnoteitemheader added	1
lateximage: pdfcrop: --hires added.	390	Introduction: MathJax support mentioned.	42
Reorganize \HomeHTMLFilename logic.	390	MathJax support added. 373, 377, 379	
Surpress extra space.	390	Options: mathsvg and mathjax	132

Supports colored <code>\rule</code>	646	<code>\LWR@subhtmlclass</code> :	
<code>titles: null \pagestyle</code> and		Factored code.	226
<code>\thispagestyle</code> for HTML. . .	609	<code>\SetHTMLFileNumber</code> : Add: Control	
<code>tikzpicture</code> : Adapts to tikz version.	608	file numbers.	216
<code>equation</code> : MathJax support.	375	<code>\cpagerefFor</code> : User-redefinable	
v0.20		word for page references.	403
<code>\BlockClassSingle</code> : Renamed from		<code>\dotfill</code> : Inserts an ellipsis.	420
" <code>LWR@htmldivclassline</code> ".	229	<code>\hfill</code> : Inserts a <code>\quad</code>	420
<code>\HTMLDescription</code> : Added		<code>\hrulefill</code> : Inserts a short rule. . .	420
<code>\NewHTMLdescription</code> .		<code>\hyperindexref</code> : Print mode	
(Renamed in v0.30.)	240	provided in case <code>hyperref</code> not	
<code>\HTMLFilename</code> : No longer escape		used.	365
underscores.	216	<code>\pageref</code> : Added.	346
<code>\HomeHTMLFilename</code> : No longer		<code>\tracinglwrap</code> : Added.	150
escape underscores.	216	General: 2017/02/09	1
<code>\InlineClass</code> : Renamed from		<code>afterpage</code> : Added.	438
" <code>inlineclass</code> ".	229	<code>alltt</code> : Added.	440
<code>\LWR@LwrapStart</code> : Fix: math cross		<code>bookmark</code> : Added.	448
references.	266	caption and subcaption supported. .	1
<code>\LWR@closeparagraph</code> : <code>\unskip</code>		<code>cleveref</code> and referencing patches:	
extra spaces.	233	Applied <code>\AfterEndPreamble</code> . .	402
No break tags in the start/end of a		<code>draftwatermark</code> : Added.	456
tabular.	233	<code>eso-pic</code> : Added.	461
<code>\LWR@endofline</code> : Fix: <code>\\</code>	422	<code>everypage</code> : Added.	461
<code>\LWR@filestart</code> : Adds meta		<code>extramarks</code> : Added.	462
<code>description</code>	262	<code>fancyhdr</code> : Added.	467
<code>\LWR@hspace</code> : Add: Supports HTML		<code>hyperref</code> : Additional user macros. .	508
thin breakable space.	423	<code>keyfloat</code> : Added.	515
<code>\LWR@htmldivclass</code> : Added		<code>letterspace</code> : User-interface	
optional style.	227	emulated.	517
<code>\LWR@htmlclass</code> : Added		<code>listings</code> : Added.	519
optional style.	227	<code>lrcaption</code> : Added.	524
<code>\LWR@htmlsectionfilename</code> :		<code>lwrap-newproject</code> : Added.	158
<code>HTMLFilename</code> : removed		<code>microtype</code> : User-interface	
additional trailing <code>'-</code> , and may be		emulated.	538
empty.	217	<code>needspace</code> : Added.	546
Sections called "Index" or "index"		<code>nowidow</code> : Added.	548
have an underscore prepended to		<code>placeins</code> : Added.	563
their filenames if no prefix. . . .	217	<code>ragged2e</code> : Added.	565
<code>\LWR@includegraphicsb</code> : Fix:		<code>setspace</code> : Improved support. . . .	579
<code>\linewidth</code> in a floatrow.	499	<code>textpos</code> : Added.	602
Fix: Expands filename.	499	<code>titles</code> : Added.	609
<code>\LWR@longtabledatacaptiontag</code> :		<code>titlesec</code> : Added.	612
Fix: Pars in captions.	324	<code>titletoc</code> : Added.	614
<code>\LWR@section</code> : Combined		<code>titling</code> : Improved compatibility. .	616
higher-level sections together		<code>tocloft</code> : Added.	624
into files.	254	<code>wallpaper</code> : Added.	640
<code>\LWR@setOSWindows</code> : Auto-detects		<code>wrapfig</code> : Added.	641
operating system.	130	Added <code>@</code> , <code><</code> , <code>></code> columns.	292

Added single-expansion data arrays.	214	lwarpmk: Fix: lwarpmk again for Windows.	199
Code factored into independent lwarpmk files.	432	lwarpmk: Fix: lwarpmk images for Windows.	199
Docs: Examples for generating HTML file names.	78	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script.	199
Docs: Improved index.	1	Add: Errors for misplaced packages.	135
Enhanced titling support.	272	Docs: Added internet class.	47
File: lwarpmk.css: Minor fixes for validation.	1	Docs: Added TeX2page, GladTeX.	47
File: lwarpmk used to compile print, HTML, indexes, and lateximages.	1	Docs: Installing on Windows.	53
Fix: \linewidth in a floatrow.	482	File lwarpmk_tutorial.txt added.	57
Improved float caption type handling.	476	v0.22	
Moved sidebar and example code to test suite.	1	\LWR@parseDcolumn: Added tabular D column.	305
Page geometry set to 6in wide with large margins.	142	\LWR@parsebangcolumn: Added tabular ! column.	302
Parallel versions of aux files for print/HTML.	1	\LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns.	307
Removed reliance on make, grep, gawk.	1	\LWR@printmccoldata: Added tabular D, !, and X columns.	320
Tabular: \unskip extra spaces.	292	General: 2017/03/02	1
Test Suite: HTML meta descriptions.	1	abstract: Added.	435
Verbatim: Added.	469	change page: Added.	452
verbatim: Added.	283	dcolumn: Added.	455
BlockClass: Added optional style.	228	ftnright: Added.	492
Renamed from "blockclass".	228	geometry: Nullified commands.	493
LWR@nestspan: Fix: Minipages inside a span.	223	indentfirst: Added.	514
v0.21		layout: Added.	516
\LWR@LwarpStart: Changed lateximages to a .txt file.	264	lscap: Added.	524
\LWR@filestart: Skip title if not given.	262	mcaption: Added.	528
\LWR@newhtmlfile: Skip title if not given.	248	nameref: Added.	546
\marginpar: Fixed source listing.	244	nextpage: Added.	547
\marginparBlock: Fixed source listing.	245	parskip: Added.	562
General: 2017/02/23	1	showkeys: Added.	581
fontenc: Added.	486	sidecap: Added.	581
fontspec: Added.	487	tabularx: Added.	597
inputenc: Added.	515	varioref: Supported.	92
newclue: Added.	546	verse: Added.	638
newunicodechar: Added.	547	v0.23	
		\LWR@parsetablecols: Fix for vert bar column type.	307
		\LWR@printmccoldata: Fix for vert bar column type.	320
		General: 2017/03/02	1
		v0.24	
		\LWR@hspace: Add: \hspace \fill converts to 2em	423

<code>\LWR@htmlfileref</code> : Fix: Index links	
while <code>\tracinglwarp</code>	342
<code>\hypertocfloat</code> : List of floats	
responds to <code>lofdepth</code> ,	
<code>lotdepth</code>	362
General: 2017/03/15	1
<code>floatrow</code> : Support for <code>subfig</code>	479
<code>subfig</code> : Added.	589
<code>tikz</code> : For <code>tikz v3.0.0</code> or later,	
auto-loads <code>tikz babel</code> library if	
necessary.	608
Docs: Filename underscore.	75, 86
No longer preloads <code>subcaption</code> ;	
conflicted with <code>subfig</code>	144
<code>picture</code> : Fix for inline images.	403
<code>tikzpicture</code> : Fix for inline images.	608
v0.25	
<code>\LWR@loadnever</code> : Added the ability	
to prevent conflicting packages.	136
<code>\addcontentsline</code> : Handles	
theorems.	355
General: 2016/03/22	1
<code>amsthm</code> : Added.	440
<code>ccaption</code> : Prevented.	452
<code>ellipsis</code> : Added.	457
<code>emptypage</code> : Added.	458
<code>framed</code> : Added.	490
<code>lips</code> : Added.	518
<code>mdframed</code> : Help avoid	
hyphenation.	530
<code>ntheorem</code> : Added.	548
<code>showidx</code> : Added.	581
<code>theorem</code> : Added.	603
Basic \LaTeX theorems: improved	
<code>css</code>	283
Docs: Adds credits for patched	
code.	1
Docs: Testing <code>lwarp</code>	119
Fix: Allows $X\TeX$ and $\text{Lua}\TeX$ to	
preload <code>graphics</code> and <code>graphicx</code>	136
v0.26	
General: 2017/03/31	1
<code>cutwin</code> : Added.	454
<code>endnotes</code> : Added.	458
<code>floatflt</code> : Added.	478
<code>footmisc</code> : Added.	487
<code>footnotehyper</code> : Added.	489
<code>footnote</code> : Added.	488
<code>marginfix</code> : Added.	527
<code>marginnote</code> : Added.	528
<code>mparhack</code> : Added.	541
<code>pagenote</code> : Supported as-is.	561
<code>sidenotes</code> : Added.	582
<code>lwarp.css</code> : Improved responsive	
<code>marginpar</code> and <code>marginblock</code>	160
Docs: Improved $\text{Mi}\TeX$ install	
instructions.	50, 53
Dollar span avoided in a	
<code>lateximage</code>	370
Footnotes now are \LaTeX boxes	
instead of <code>pagenotes</code>	240
<code>lateximage</code> : Labels track page	
numbers of <code>lateximages</code>	390
Print mode now uses a <code>minipage</code>	
of <code>\linewidth</code>	390
<code>picture</code> : Fix for <code>\makebox</code> in	
<code>picture</code>	403
v0.27	
<code>\LWR@footnotetext</code> : Fix for table	
footnote par tags.	241
General: 2017/04/04	1
<code>lettrine</code> : Added.	517
<code>microtype</code> : Fix with $\text{Xe}\TeX$,	
$\text{Lua}\TeX$	538
<code>soul</code> : Added.	585
<code>ulem</code> : Added.	635
Docs: Installing utilities for	
MacOS.	55
Docs: Limitations of <code>saveboxes</code>	87
Page geometry modified to reduce	
line overflow.	142
v0.28	
<code>\@wrindex</code> : Improved indexing.	364
<code>\HTMLAuthor</code> : Added <code>\HTMLauthor</code> .	
(Renamed in v0.30.)	239
<code>\LWR@LwarpEnd</code> : If <code>FormatEpub</code> or	
<code>FormatWordProcessor</code> , no	
bottom nav.	267
<code>\LWR@LwarpStart</code> :	
<code>FormatWordProcessor</code> forces	
single-file output.	264
<code>\LWR@filestart</code> : Adds HTML meta	
author.	262
<code>\LWR@forcenewpage</code> : Forces new	
PDF page before major	
environments.	220
<code>\LWR@htmlcomment</code> : Breaks ligatures	
in HTML comments.	226
<code>\LWR@includegraphicsb</code> : Adapts to	
<code>graphics</code> syntax.	499

\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav.	248	\HTMLDescription: Renamed from \NewHTMLdescription.	240
\LWR@parsetablecols: Added L, C, R, J column types.	307	\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	237
\LWR@startref: Removed space. . .	344	\HTMLLanguage: Renamed from \MetaLanguage.	261
\chapter: If EPUB, prints footnotes before each section.	260	\HTMLPageBottom: Renamed from \SetPageBottom.	238
\hyperindexref: Improved indexing.	365	\HTMLPageTop: Renamed from \SetPageTop.	237
\textup: Fixed span class.	416	General: 2017/04/29	1
General: 2017/04/14	1	lwrap-newproject removed, and combined with lwrap.	158
glossaries: Added.	494	lwrapmk: Add: xdyfile configuration option.	199
graphics: Added.	495	lwrapmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex.	199
tabularx: Fix for optional pos. . .	597	lwrapmk: Fix: xindy now used for print index generation with latexmk.	199
tabulary: Added.	597	lwrapmk: language now used for both index and glossary generation.	199
lwrapmk: Add: printglossary and htmlglossary commands.	199	File: lwrap_html.xdy renamed to lwrap.xdy.	197
Added boolean FormatEPUB.	151	Fix: *.css files only written in print mode.	160
Added boolean FormatWP.	151	Fix: lwrap.xdy only written in print mode.	197
Added boolean HTMLDebugComments.	150	Fix: lwrap_mathjax.txt: Only written in print mode.	197
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42.	152	Option OSWindows replaces macro \warpOSWindows.	133
Docs: Modfying lwrapmk and index processing.	119	Option latexmk replaces macro \UseLatexmk.	133
File lwrap_mathjax.txt: Updated CDN repository.	197	Option lwrapmklang changed to IndexLanguage.	132
Forced oneside to maintain large right margin.	142	Option xdyFilename added.	132
v0.29		Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename.	133
\LWR@includegraphicsb: Fix: Error when no optional arguments.	499	v0.31	
General: 2017/04/15	1	General: 2017/05/15	1
lwrapmk: Add: language option for config files.	199	keyfloat: Improved compatibility.	515
Add: lwrapmklang option for lwrap.	132	v0.32	
Docs: Using a glossary	68	\RequirePackage: Fix: Ignores blanks in package list.	146
File *.lwrapmkconf: Add: language option for config files.	159	General: 2016/06/09	1
File lwrapmk.conf: Add: language option for config files.	158		
v0.30			
\CSSFilename: Renamed from \NewCSS.	238		
\HTMLAuthor: Renamed from \HTMLauthor.	239		

glossaries: Prevent error with <code>\glo@name</code> not defined.	365	<code>\LWR@htmlspanclass</code> : Moved optional argument in front of mandatory.	224
<code>\lwarpmk</code> : Fix: <code>io.lines()</code> changed to <code>file:lines()</code> due to luatex changes.	199	<code>\LWR@nullfonts</code> : Improved font control.	417
v0.33		<code>\LWR@restoreorigformatting</code> : booktabs: Works inside <code>lateximage</code>	365
<code>\HTMLAuthor</code> : Fix: Provides empty default author if none given. . .	239	Improved font control.	365
<code>\LWR@loadbefore</code> : Fix: No <code>\PackageError</code> if already loaded.	135	<code>\LWR@subhtmlclass</code> : Moved optional argument in front of mandatory.	226
<code>\LWR@parseatcolumn</code> : Fix: Column alignment with leftmost @. . . .	301	<code>\LWR@tabledatacolumnntag</code> : booktabs: Works inside <code>lateximage</code>	331
<code>\LWR@tabledatasinglecolumnntag</code> : Fix: Macros in tabular could cause extra data cell.	312	<code>\fboxBlock</code> : Added.	412
<code>\LWR@vspace</code> : Add: <code>\vspace</code> nullified.	425	<code>\makebox</code> : Fix: Handles paren arg. .	410
<code>\StartDefiningTabulars</code> : Add: Avoids error: misplaced alignment tab character &.	296	General: 2017/08/08	1
General: 2017/07/10	1	babel-french: Adds fixed-width HTML spaces to punctuation. . .	222
<code>amsmath</code> : Removed <code>fleqn</code> option. . . .	144	balance: Added.	446
<code>fancyhdr</code> : Fix: Optional args for <code>\thead</code> , etc.	467	booktabs: Works inside <code>lateximage</code>	334, 449
Add: Tabular at and bang columns now have their own HTML columns.	292	<code>boxedminipage2e</code> : Added.	449
<code>cleveref</code> : Fix: Loaded <code>\AtEndPreamble</code>	402	<code>boxedminipage</code> : Prevented.	449
Fix: Incorrectly-inline math environments.	379	<code>crop</code> : Added.	453
New handling of & to localize catcode changes.	292	<code>enumerate</code> : Added.	459
v0.34		<code>enumitem</code> : Added, no longer required.	459
<code>@fnsymbol</code> : Text symbols instead of math.	275	<code>everyshi</code> : Added.	462
<code>\InlineClass</code> : Moved optional argument in front of mandatory. . . .	229	<code>fancybox</code> : Added.	464
<code>\LWR@htmldivclass</code> : Moved optional argument in front of mandatory.	227	<code>fancyvrb</code> : Added, no longer required.	469
<code>\LWR@htmlclass</code> : Moved optional argument in front of mandatory.	227	<code>figcaps</code> : Added.	474
<code>\LWR@htmlclassline</code> : Moved optional argument in front of mandatory.	227	<code>filecontents</code> : Required. Patched for morewrites.	143
		<code>floatpag</code> : Added.	479
		<code>flushend</code> : Added.	485
		<code>fullpage</code> : Added.	492
		<code>hyperxmp</code> : Added.	513
		<code>idxlayout</code> : Added.	513
		<code>marginfit</code> : Added.	527
		<code>mdframed</code> : Improved <code>mdtheorem</code> patch.	535
		<code>moreverb</code> : Added.	539
		<code>paralist</code> : Added.	561
		<code>pdfscape</code> : Added.	562
		<code>pdfsync</code> : Added.	562
		<code>prelim2e</code> : Added.	563
		<code>rotfloat</code> : Added.	569

savetrees: Added.	570	\LWR@restoreorigformatting:	
shadow: Added.	580	siunitx: Improved	
syntonly: Added.	596	super/subscripts in a	
titles: No longer required.	609	lateximage.	365
titleref: Prevented.	612	\LWR@section: Improved spacing.	254
xmpincl: Added.	654	\LWR@stoppars: Extra HTML source	
Added.	649	space after paragraphs.	235
Docs: Horizontal space		\fbox: Fix: Uses \fboxrule and	
limitations.	1	\fboxsep.	411
Docs: Misplaced alignment		\framebox: Fix: Handles width and	
character.	121	horiz position.	410
File: README.txt: updated.	1	\makebox: Fix: Handles width and	
File: lwarp_mathjax.txt: Version		horiz position.	410
change.	197	General: 2017/08/17	1
Fix: Added the eqnarray		babel-french: Adjustements for	
environments.	379	French variants, load order,	
Improved font control.	415	footnotes, ellipses.	222
Lists refactored to remove		footnote: Extra HTML source space	
enumitem requirement.	285	after paragraphs.	488
Verbatim refactored to remove		siunitx: Fix for babel-french.	396
fancyvrb requirement.	281	siunitx: Improved symbol support.	584
lateximage: Fix: lateximage with		transparent: Added.	633
minipage, \parbox, \makebox,		upref: Added.	637
\fbox, \framebox, \raisebox,		xcolor: Added \fcolorboxBlock,	
\scalebox, \reflectbox.	390	\colorboxBlock.	642
BlockClass: Moved optional		xcolor: Fix: Background none in	
argument in front of mandatory.	228	print mode.	642
fminipage: Added.	412	xcolor: Refactored	
LWR@nestspan: Fix: Minipages,		\LWR@colorstyle.	646
BlocksClass, and lists inside a		xcolor: Uses \fboxrule and	
span.	223	\fboxsep.	642
LWR@tabular: booktabs: Works		xcolor: \fcolorbox etc. now work	
inside lateximage.	335	inside lateximage.	642
v0.35		Docs: Reorganized: Special cases	
General: 2017/08/08	1	and limitations.	85
Fix: \textbf and related.	415	Source: Improved formatting.	1
v0.36		lateximage: Footnotes appear in	
\LWR@HTMLsanitize: Fix for		regular text instead of the	
babel-french.	387	lateximage minipage.	390
\LWR@HTMLsanitizeexpand: Fix for		LWR@tabular: Fix for babel-french.	335
babel-french.	388	v0.37	
\LWR@closeparagraph: Extra HTML		\include: Maintains independent	
source space after paragraphs.	233	.aux files for HTML.	148
\LWR@currenttextcolor: Fix for		General: 2017/08/19	1
\rule when xcolor not loaded.	425	\LaTeX accents: Added.	156
\LWR@footnotetext: Extra HTML		babel-french: Adjustment for load	
source space after paragraphs.	241	order.	222
Force HTML superscripts.	241	color: Prevented.	453
\LWR@nullfonts: Fix: Filenames		comment: Maintains independent	
while using MathJax.	417	cutfiles for print, HTML.	134

siunitx: Improved symbol support.	584		
textcomp: Improved support. . . .	599		
lwarpmk: Removes additional HTML aux files.	199		
File handles reorganized.	147		
v0.38			
\@secntformat: Added for appendix.	254		
\ForceHTMLPage: Added.	251		
\ForceHTMLTOC: Added.	252		
\LWR@section: \part* starts a new HTML page, for appendix.	254		
Modified spacing, uses \numberline.	254		
\numberline: Added trailing \quad.	361		
\part: Fix with article class. . . .	259		
General: 2017/08/27	1		
appendix: Added.	444		
arabicfront: Added.	444		
caption2: Prevented.	451		
chappg: Added.	453		
color: Forces xcolor as well. . . .	453		
fix2col: Added.	474		
fncychap: Added.	485		
grffile: Added.	507		
metalogo: Added.	538		
nonumonpart: Added.	547		
nopageno: Added.	548		
pagenote: Option page disabled.	561		
realscripts: Added.	566		
relsize: Added.	567		
romanbarpagenumber: Added. . .	569		
romanbar: Added.	568		
scalefnt: Added.	571		
siunitx: Removed from lwrap core.	584		
textcomp: Removed from lwrap core.	599		
tocbibind: Added.	622		
xltxtra: Added.	653		
lwarpmk: Added print1 and html1 actions.	199		
Added \markboth, \sloppy, etc.	220		
Docs: Enhanced <i>Supported Functions</i> table.	43		
Docs: Index, tocbibind.	95		
Docs: Starred sections.	92		
v0.39			
\@maketitle: titling version.	619		
Native \LaTeX version.	276		
Removed minipages.	276, 619		
		Supports authblk with <div>s of class oneauthor instead of tabular.	276, 619
		\AddSubtitlePublished: Added. . .	277
		\LWR@domulticolumn: Add: Optional vpos and # rows.	322
		\LWR@restoreorigformatting: Appended with \appto instead of calling various macros.	365
		\LWR@tabledatacolumnntag: Don't start a data cell if see \TabularMacro.	331
		\ResumeTabular: Added.	330
		\TabularMacro: Added.	330
		\multicolumnrow: Added.	328, 545
		\printauthor: Removed minipages.	272
		Supports authblk with <div>s of class oneauthor instead of tabular.	272
		\thanksmarkseries: Removed minipage footnotes.	620
		General: 2017/09/05	1
		a4wide: Added.	435
		a4: Added.	434
		a5comb: Added.	435
		addlines: Added.	438
		anysize: Added.	443
		authblk: Added.	445
		bigdelim: Added.	447
		bigstrut: Added.	448
		chngepage: Prevented.	452
		ebook: Added.	457
		fullwidth: Added.	493
		supertabular: Added.	594
		textarea: Added.	598
		titling: Improved compatibility. .	616
		titling: Removed extraneous center environments.	617
		typearea: Added.	635
		xtabular: Added.	654
		zwpagelayout: Added.	656
		Add: New optional vpos argument.	543
		Add: Supports left/right border for bigdelim.	543
		Docs: Reorganized tabular discussion.	102
		Fix: Long text argument.	543
		Titlepage \published and \subtitle removed.	

\AddSubtitlePublished		quotchap: Added.	564
restores.	276	sectsty: Added.	578
titlepage: Clear pending footnotes.	271	stabular: Added.	588
Removed minipages.	271	tabs: Added.	596
titlingpage: Clear pending		textcomp: Additional symbols,	
footnotes.	617	improved XeLaTeX and LuaLaTeX	
v0.40		support.	599
\@chapcntformat: Added for		tocbibind: Improved for	
tocbibind, anonchap.	254	\simplechapter.	622
\LWR@hline: Added.	334	xltxtra: Fix for \showhyphens with	
\LWR@includegraphicsb: Add: Full		XeLaTeX.	653
\graphicspath support.	499	Improved bigdelim borders.	543
\LWR@nullfonts: Fix: Long		No longer preloads xfrac.	144
arguments for expandable		v0.41	
command.	417	\LWR@addcmidruletrim: Add:	
\LWR@restoreorigformatting:		\cmidrule trims.	316
Improved \TeX logos inside a		\LWR@clearmidrules: Add:	
lateximage.	365	\cmidrule trims.	314
Improved symbols inside a		\LWR@closetabledatacell: Add:	
lateximage.	365	Mute > for \bottomrule.	294
Nullified \InlineClass, etc.		Fix: At/bang column with	
inside a lateximage.	365	\multirow.	295
\LWR@tabledatacolumntag: Fix for		Fix: Cancel < for \multicolumn.	295
bigdelim: \ldelim, \rdelim.	331	\LWR@domulticolumn: Add:	
\chapter: Added support for		\cmidrule trims.	322
quotchap.	260	Added vertical rules.	323
\multicolumnrow: Fix: Adapts to		\LWR@nullifyNoAutoSpacing: Fix:	
older multirow and xparse.	328	\NoAutoSpacing in a tabular	
\simplechapterdelim: Added for		with babel-french.	334
tocbibind, anonchap.	254	\LWR@parsebarcolumn: Added	
\underline: Added.	420	vertical rules.	304
General: 2017/09/25	1	\LWR@printatbang: Add: \cmidrule	
adjmulticol: Added.	437	trims.	310
anonchap: Added.	443	Add: Mute at and bang columns	
bigdelim: Improved		for \bottomrule.	310
documentation.	447	\LWR@printbartag: Added vertical	
cuted: Added.	454	rules.	310
dblfnote: Added.	455	\LWR@subaddcmidruletrim: Added.	316
fnpos: Added.	486	\LWR@subcmidrule: Add: \cmidrule	
graphics: Moved out of the lwarp		trims.	315
core.	495	\LWR@tabledatasinglecolumntag:	
graphics: Restores		Add: \cmidrule trims.	312
\includegraphics and		Add: Mute < for \bottomrule.	312
\DeclareGraphicsExtensions		\LWR@tabularfinishrow:	
in a lateximage.	495	Unfinished tabular rows	
graphicx: Moved out of the lwarp		automatically filled.	297
core.	506	\mcolrowcell: Added for	
grffile: Directly supported.	507	\multicolumnrow cells.	333
midfloat: Added.	539	General: 2017/10/07	1
pfnote: Added.	563	Add: \cmidrule trims.	543

Added vertical rules.	544	\LWR@singledollar: If FormatWP print LaTeX expression.	371
Fix: < spec.	544	\LWR@subaddcmidruletrim: Opt if no rule given.	316
\LWR@tabular: Fix: \NoAutoSpacing in a tabular with babel-french.	335	\LWR@tabledatasinglecolumn: If FormatWP add cell alignment.	313
Improved rules.	336	\LaTeX: If FormatWP use explicit style.	429
v0.42		\TeX: If FormatWP use explicit style.	429
\@ensuredmath: Improved		\listoffigures: Added boolean WPMarkLOFT.	358
\ensuremath.	372	\listoftables: Added boolean WPMarkLOFT.	358
\@textsubscript: Added.	419	\marginpar: If FormatWP emulate a wrapfig.	244
\@textsuperscript: Added.	419	\marginparBlock: If FormatWP emulate a wrapfig.	245
\LWR@HTMLtextstyle: Added.	415	\tableofcontents: Added boolean WPMarkTOC.	358
\LWR@addformatwpalignment: If FormatWP add explicit style for cell alignment.	318	\underline: If FormatWP, use explicit styles for \underline, etc.	420
\LWR@addrulewidth: If FormatWP force explicit border.	317	General: 2017/10/30	1
\LWR@amsmathbody: Fix: Numbering and naming AMS math environments.	390	\textbf and related: If FormatWP, use explicit styles for \textsc, etc.	415
\LWR@amsmathbodynumbered: Fix: Numbering and naming AMS math environments.	390	algorithmicx: If FormatWP add \quads.	439
\LWR@domulticolumn: If FormatWP add cell alignment.	323	epigraph: If FormatWP add HTML styles.	460
\LWR@doubledollar: If FormatWP print LaTeX expression.	371	fancybox: If FormatWP add HTML styles.	464
Improved \ensuremath.	371	floatflt: Added width.	478
Improved line spacing with mathjax.	371	includegraphics: Fix: Class key.	499
\LWR@floatbegin: If FormatWP add a text frame.	350	keyfloat: If FormatWP add explicit HTML style.	516
\LWR@floatend: If FormatWP add a text frame.	351	moreverb: Simplified formatting of listings.	539
\LWR@hline: If FormatWP force explicit border.	334	morewrites: Added.	541
\LWR@hspace: If FormatWP add \quads.	424	multirow: If FormatWP add cell alignment.	544
\LWR@htmlmathlabel: If FormatWP print LaTeX expression.	377	overpic: Added.	560
\LWR@includegraphicsb: Fix: Filename expansion.	499	realscripts: Fix for subscripts in a lateximage.	566
If FormatWP, use explicit size.	501	sidenotes: If FormatWP add explicit HTML style.	583
\LWR@remembertag: Fix: Numbering and naming AMS math environments.	389	siunitx: Improved \ensuremath.	584
\LWR@restoreorigformatting: Improved \ensuremath.	367	soul: If FormatWP, add explicit styles.	585
\LWR@rule: If FormatWP add \quads.	427		

textcomp: Improved	
\interrobangdown.	599
wrapfig: If FormatWP add explicit HTML style.	641
Added boolean WPMarkLOFT.	152
Added boolean WPMarkMath.	152
Added boolean WPMarkMinipages.	152
Added boolean WPMarkTOC.	152
Added boolean WPTitleHeading.	153
Docs: Added support page.	2
Docs: Improper \prevdepth.	121
Docs: Reorganized math limitations	96
File: lwarp_mathjax.txt: Updated siunitx script.	197
Fix: Numbering and naming AMS math environments.	388
If FormatWP, shift section headings.	153
tabbing: Added.	283
lateximage: Fix: Numbering and naming AMS math environments.	391
center: If FormatWP use explicit text-align.	394
minipage: Added boolean WPMarkMinipages.	408, 409
If FormatWP add a text frame.	407
eqnarray: Fix: Numbering and naming AMS math environments.	380
If FormatWP print LaTeX expression.	379
equation: If FormatWP print LaTeX expression.	376
LWR@BlockClassWP: Added to factor code.	229
LWR@figcaption: If FormatWP forces italic captions.	353
LWR@tabular: If FormatWP force explicit border.	336
v0.43	
\LWR@domulticolumn: Fix for vertical rules.	323
Fix: Multicolumn trim.	322, 323
\LWR@maybeprintpendingfootnotes: Added FootnoteDepth.	244
\LWR@nullfonts: Fix: Nullify dollar inside filenames.	417
\LWR@parsetablecols: Ignore spaces in col spec.	307
\LWR@printmccoltype: Added vertical rules.	319
\LWR@restoreorigformatting: Fix: \ref in math or lateximage.	365
Fix: Improved math inside lateximages.	367
\LWR@section: Fix: Expansion in comparison.	255
Fix: Math in section name.	256, 258
Fix: Nullify fonts inside HTML comment.	256
\TabularMacro: \newcommand instead of \relax to fix supertabular and xtab.	330
\href: Made robust.	347
\nameref: Made robust.	346
\nolinkurl: Made robust.	347
\url: Made robust.	348
General: 2017/11/08	1
breakurl: Added.	450
hyperref: Made robust.	509–511
hyperref: \Gauge added.	512
luatodonotes: Added.	525
todonotes: Added.	631
LWR@currentautosec: Added.	254
Added FootnoteDepth.	241
Docs: HTML settings table.	74
Docs: Reorganized HTML customization.	74
v0.44	
\@currentlabelname: Adjustment for koma-script.	339
\HTMLTitle: Added.	239
\LWR@addformatwpalignment: Fix for multicolumn alignment if FormatWP.	318
\LWR@backgroundcolor: Added.	646
\LWR@filestart: Add \HTMLTitle.	263
Fix \HTMLAuthor.	262
\LWR@listitem: Added list and trivlist.	285
\LWR@patchlists: Added list and trivlist.	288
\LWR@strresult: Fix: \providecommand.	300
\addcontentsline: Automatic \LWR@newfloatanchor.	355

<code>\chapter</code> : Add preamble for koma-script.	260	<code>sclayer-scrpage</code> : Added.	576
<code>\marginparBlock</code> : Added.	245	<code>sclayer</code> : Added.	574
<code>\nopagecolor</code> : xcolor: Fix for <code>\nopagecolor</code>	647	<code>section</code> : Added.	577
<code>\part</code> : Add preamble for koma-script.	259	<code>soulpos</code> : Added.	587
<code>\title</code> : Added <code>\thetitle</code>	239	<code>soulutf8</code> : Added.	588
General: 2017/11/22	1	<code>supertabular</code> : Fix for caption.	595
AMS environments: Fix: Groups for lateximages.	381	<code>tocbasic</code> : Added.	621
algorithmicx: Improved comment symbol.	439	<code>tocloft</code> : Added <code>\newlistentry</code>	628
atbegshi: Added.	445	<code>tocloft</code> : Improved <code>\newlistof</code>	629
cancel: Added.	450	<code>tocstyle</code> : Added.	630
changepage: Additional options.	452	<code>todonotes</code> : Improved.	631
easy-todo: Added.	456	<code>todo</code> : Added.	630
fancyref: Added.	467	<code>typearea</code> : Added expert commands.	635
fixmetodonotes: Added.	476	<code>watermark</code> : Added.	640
fixme: Added.	475	<code>xcolor</code> : Added <code>\LWR@currenttextcolorstyle</code>	645
fontenc: Allowed after <code>\lwrap</code>	486	<code>xcolor</code> : Added <code>\LWR@findcurrentcolor</code>	645
hang: Added.	507	<code>xcolor</code> : Added <code>\LWR@textcurrentcolor</code>	646
ifoddpage: Added.	514	<code>xtab</code> : Fix for caption.	655
ltxtable: Added.	525	Adjustment for koma-script.	140, 142
luatodonotes: Improved.	525	If pdfLaTeX, require T1 and UTF8 encoding.	127
<code>\lwrap-patch-komascript</code> : Added.	433	<code>picture</code> : <code>overpic</code> : Fix: Groups for lateximages.	403
<code>overpic</code> : Fix: Groups for lateximages.	560	<code>tikzpicture</code> : <code>tikz</code> : Fix: Groups for lateximages.	608
<code>pdfsync</code> : Fixes.	562	<code>list</code> : Added <code>list</code> and <code>trivlist</code>	285
<code>preview</code> : Added.	564	<code>LWR@nestspan</code> : Added <code>list</code> and <code>trivlist</code>	223
<code>preview</code> : Added.	564		
<code>scnextend</code> : Added.	571		
<code>scrhack</code> : Added.	574		
<code>sclayer-notecolumn</code> : Added.	576		

§ 277 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			
<code>\\$</code>	<i>370</i>	<code>\@@setcref</code>	<u>7476</u>
<code>\&</code>	<i>215, 5126</i>	<code>\@@setcrefrange</code>	<u>7477</u>
<code>\(</code>	<u>6754</u>	<code>\@author</code>	<i>269</i>
<code>\)</code>	<u>6754</u>	<code>\@begintheorem</code>	<u>4925</u>
<code>\,</code>	<i>85</i>	<code>\@capttype</code>	<u>6322</u>
<code>\@@setcpageref</code>	<u>7480</u>	<code>\@chapcntformat</code>	<u>4279</u>
		<code>\@currentlabelname</code>	<u>6093</u>

<code>\@date</code>	269	<code>\AmS</code>	8096
<code>\@dlbfloat</code>	6294	AMSMath	
<code>\@endtheorem</code>	4935	split miss-numbered	99, 549
<code>\@ensuredmath</code>	6762	amsmath (package)	144
<code>\@float</code>	6294	amsthm (package)	440
<code>\@fnsymbol</code>	4734	<code>\and</code>	269
<code>\@footnotetext</code>	4020	anonchap (package)	443
<code>\@include</code>	490	anysize (package)	443
<code>\@makecaption</code>	6324	appendix (package)	444
<code>\@makefnmark</code>	3991	arabicfront (package)	444
<code>\@makefntext</code>	3990	array (package)	339
<code>\@maketitle</code>	49, 4757	AsciiDoc (program)	48
<code>\@mpfootnotetext</code>	4021	AsciiDoctor (program)	48
<code>\@opargbegintheorem</code>	4930	Asciidoctor-LaTeX (program)	48
<code>\@secntformat</code>	4277	atbegshi (package)	445
<code>\@starttoc</code>	6435	<code>\attrib</code>	107, 279, 638
<code>\@textsubscript</code>	7867	<code>\attribution</code>	4830
<code>\@textsuperscript</code>	7863	authblk (package)	445
<code>\@title</code>	269	author	
<code>\@wrglossary</code>	6603	HTML meta tag	84, 239
<code>\@wrindex</code>	6594	<code>\author</code>	83, 269
<code>\\</code>	422	autosec	253
<code>\$</code>	371		
<code>\$\$</code>	371		
<code>~</code>	85		
		B	
	A	babel (package)	108
a4 (package)	434	<code>\backmatter</code>	4261
a4wide (package)	435	balance (package)	446
a5comb (package)	435	<code>\BaseJobname</code>	3401
abstract (environment)	4820	BaseJobname (option)	73, 132
abstract (package)	435	baseline	
<code>\abstractname</code>	76, 4819	tabular	306
accents	251	<code>\bfseries</code>	7852
<code>\addcontentsline</code>	6386	<code>\BibTeX</code>	8091
addlines (package)	438	bigdelim (package)	447
<code>\AddSubtitlePublished</code>	4782	bigstrut (package)	448
adjmulticol (package)	437	BlockClass (environment)	3721
Adobe (program)	49	<code>\BlockClassSingle</code>	3733
<code>\affiliation</code>	4644	bookmark (package)	448
afterpackage (package)	142	booktabs (package)	449
afterpage (package)	438	booleans:	
algorithmic		CombineHigherDepths	76, 246
with newfloat, trivfloat	634	FileSectionNames	76, 216
algorithmicx (package)	439	FormatEPUB	110, 151
align (environment)	7029	FormatWP	112, 151
align* (environment)	7052	HTMLDebugComments	150
alignment tab character &, misplaced		LWR@amsmultline	377
	102, 289, 296	LWR@doingapar	231
alltt (package)	440	LWR@doingcmidrule	293
		LWR@doinghline	293
		LWR@doingstartpars	231

LWR@doingtbrule	293	\centering	265
LWR@emptyatbang	294	change page (package)	452
LWR@exitingtabular	331	chappg (package)	453
LWR@freezethisautoid	351	\chapter	4427
LWR@infloatrow	499	chnpage (package)	452
LWR@intabularmetadata	294	\citetitle	4837
LWR@isstartingequation	389	class:	
LWR@minipagefullwidth	406	internet	47
LWR@minipagethispar	406	\cleardoublepage	7992
LWR@skipatbang	294	\clearpage	7992
LWR@skippingmcolrowcell	293	cleveref (package)	401
LWR@skippingmrowcell	293	cmap (package)	71
LWR@starredlongtable	324	\color	647
LWR@startedrow	293	color (package)	453
LWR@tableparcell	293	\colorbox	72
LWR@tabularcelladded	293	\colorboxBlock	82, 7372
LWR@tabularmutemods	331	CombineHigherDepths (boolean)	76, 246
LWR@tracinglwarp	150	comment (package)	134
LWR@validtablecol	305	Computer Modern	70
LWR@verbtags	281	\ConTeXt	8088
mathjax	131	counters:	
usingOSWindows	130	FileDepth	76, 246
warpingHTML	131	FootnoteDepth	76, 241
warpingprint	131	lofdepth	361
WPMarkFloats	114, 151	lotdepth	362
WPMarkLOFT	115, 152	LWR@currentautosec	254
WPMarkMath	115, 152	LWR@externalfilecnt	370
WPMarkMinipages	114, 152	LWR@htmlfilenumber	217
WPMarkTOC	114, 152	LWR@latestautopage	353
WPTitleHeading	115, 153	LWR@lateximagedepth	386
boxedminipage (package)	449	LWR@lateximagenumber	386
boxedminipage2e (package)	449	LWR@LIpage	386
\boxframe	164	LWR@midrulecounter	314
breakurl (package)	450	LWR@minipagedepth	404
bugs	121	LWR@nextautoid	352
BVerbatim (environment)	169	LWR@nextautopage	352
		LWR@nextequation	373
		LWR@prevFileDepth	253
		LWR@spandepth	231
		LWR@startingequation	388
		LWR@tablecolindex	301
		LWR@tablecolspecindex	300
		LWR@tablecolspecwidth	300
		LWR@tabletotalcols	301
		LWR@tabletotalcolsnext	301
		LWR@thisautoid	351
		SideTOCDepth	75, 359
		tocdepth	75
		\cpagerefFor	7479
		crop (package)	453
C			
calc (package)	143		
Calibre	110		
cancel (package)	450		
caption (package)	144		
caption2 (package)	451		
\caption@begin	6367		
\caption@end	6367		
\captionlistentry	6371		
\captionof	6400		
\CaptionSeparator	6323		
ccaption (package)	452		
center (environment)	7313		

cross-references	
missing or incorrect	121
CSS	
file selection	79
lwrap.css	80
per HTML page	79
project-specific changes	80
\CSSFilename	77, 79, 3942
cuted (package)	454
cutwin (package)	454
D	
danger icon	126
\date	83
dblfnote (package)	455
dcolumn (package)	455
debugging	121
HTML debug comments	150
\DeclareGraphicsExtensions	2
Deja Vu	70
description	
HTML meta tag	83, 240
description (environment)	5002
displaymath (environment)	6776
\dotfill	7879
draftwatermark (package)	456
E	
easy-todo (package)	456
ebook (package)	457
ellipsis (package)	457
\emph	7752
emptypage (package)	458
\end@dblfloat	6310
\end@float	6310
\EndDefiningTabulars	5120
endnotes (package)	458
\enlargethispage	7991
\enskip	422, 7944
enumerate (environment)	4983
enumerate (package)	459
enumitem (package)	459
environ (package)	144
environments:	
abstract	4820
align	7029
align*	7052
BlockClass	3721
BVerbatim	169
center	7313
description	5002
displaymath	6776
enumerate	4983
eqnarray	6884
equation	6823
equation*	6847
fcolorminipage	148, 7414
flalign	7075
flalign*	7098
flushleft	7329
flushright	7321
fminipage	88, 7695, 7743
gather	6983
gather*	7006
itemize	4978
lateximage	385, 7231
list	4949
longtable	2
LWR@BlockClassWP	3749
LWR@figcaption	6333
LWR@nestspar	3577
LWR@tabular	5973
LWR@print@fminipage	7719
math	6775
minipage	7545
multiline	6934
multiline*	6958
picture	403, 7500
quote	4845
tabbing	4913
theindex	6572
tikzpicture	8
titlepage	82, 4661
titlingpage	14, 83
Verbatim	13
verbatim	4903
verse	2
warpall	81, 104
warpHTML	54, 78, 81, 105
warpprint	78, 81, 105
epigraph (package)	460
Epub	
conversion software	110
HTML conversion settings	110, 151
eqnarray (environment)	6884
equation	
miss-numbered	99, 549
equation (environment)	6823
equation* (environment)	6847
error messages	121

<code>\fup</code>	7871	<code>title</code>	84, 239
<code>\fussy</code>	3498	<code>viewport</code>	263
G			
<code>gather</code> (environment)	6983	<code>tabular column conversion</code>	306
<code>gather*</code> (environment)	7006	<code>\HTMLAuthor</code>	77, 84, 3972
<code>GELLMU</code> (program)	47	<code>HTMLDebugComments</code> (boolean)	150
<code>generator</code>		<code>\HTMLDescription</code>	77, 83, 3977
HTML meta tag	262	<code>\HTMLentity</code>	3384
<code>geometry</code> (package)	141, 493	<code>\HTMLFilename</code>	3402
<code>getttitlestring</code> (package)	142	<code>HTMLFilename</code> (option)	73, 75, 133
<code>GladTeX</code> (program)	48	<code>\HTMLFirstPageTop</code>	77, 3919
<code>glossaries</code> (package)	94, 494	<code>htmlglossary</code> (option) [lwarpmk] .	95, 494
<code>glossary</code>		<code>\HTMLLanguage</code>	77, 4492
language	68	<code>HTMLleftmargini</code> (length) . .	108, 280, 638
processing	68	<code>\HTMLPageBottom</code>	77, 270, 3935
<code>glyptounicode</code> (file)	71	<code>\HTMLPageTop</code>	77, 3927
<code>graphics</code> (package)	495	<code>\HTMLTitle</code>	77, 84, 3967
<code>graphicx</code> (package)	506	<code>\HTMLunicode</code>	3392
<code>grffile</code> (package)	506	<code>HTMLvleftskip</code> (length) . . .	108, 280, 638
H			
<code>hang</code> (package)	507	<code>\hyperindexref</code>	6610
<code>heading</code> , word processor	115	<code>hyperref</code> (package)	346, 508
<code>Hevea</code> (program)	47	<code>\hypertoc</code>	6518
<code>\hfill</code>	7877	<code>\hypertocfloat</code>	6531
<code>\HomeHTMLFilename</code>	3403	<code>hyperxmp</code> (package)	513
<code>HomeHTMLFilename</code> (option) . . .	73, 75, 133	I	
<code>horizontal space</code>	85	<code>icon</code>	
between minipages	422	warning	126
<code>\href</code>	6236	<code>idxlayout</code> (package)	513
<code>\hrulefill</code>	7878	<code>ifoddpages</code> (package)	514
<code>\hskip</code>	85	<code>ifplatform</code> (package)	129
<code>\hspace</code>	85, 422, 7982	<code>images</code>	
<code>htlatex</code> (program)	47	graphicx package	396
HTML		in strange places	122
commands for a successful conver-		showing as HTML	63, 122
sion	85	<code>\include</code>	490
conversion settings	74	<code>\includegraphics</code>	228
debug comments	150	<code>indentfirst</code> (package)	514
Epub	110, 151	<code>InDesign</code> (program)	49
word processor	112, 151, 152	<code>index</code>	
filename generation	78	language	68
headings	125	placement and toc options	95
in place of SVG images	63, 122	processing	61, 62
meta tag		UTF-8	71
author	84, 239	<code>IndexLanguage</code> (option)	68, 73, 95, 132, 494, 621
description	83, 240	<code>\InlineClass</code>	3741
generator	262	<code>inputenc</code> (package)	71, 515
Title	239	<code>internet</code> (class)	47
		<code>item</code>	
		empty	86, 284

lwarpmk (option)	73, 132	\LWR@endsubminipage	7530
lwarpmk (program)	119, 199	\LWR@ensuredoingapar	3808
lwarpmk.conf (file)	158	\LWR@equationtag	7190
lwarpmk.lua (file)	119	LWR@exitingtabular (boolean)	331
\LWR@addcmidruletrim	5554	LWR@externalfilecnt (counter)	370
\LWR@addcmidrulewidth	5579	LWR@figcaption (environment)	6333
\LWR@addformatwppalignment	5586	\LWR@filenamenoblanks	4106
\LWR@addleftmostbartag	5427	\LWR@filestart	4499
\LWR@addmathjax	6812	\LWR@findword	419
\LWR@addrulewidth	5559	\LWR@floatbegin	6273
\LWR@afterendverbatim	4887	\LWR@floatend	6296
\LWR@amsmathbody	7218	\LWR@footnotetext	3992
\LWR@amsmathbodynumbered	7224	\LWR@forceminwidth	7662
LWR@amsmultline (boolean)	377	\LWR@forcenewpage	3488
\LWR@atbeginverbatim	4872	LWR@freeze>thisautoid (boolean)	351
\LWR@backgroundcolor	48	\LWR@futurenonspacet	5043
\LWR@beginhideamsmath	6869	\LWR@FVstyle	17
\LWR@blackborderpadding	7673	\LWR@getexparray	3380
LWR@BlockClassWP (environment)	3749	\LWR@getmynexttoken	5050
\LWR@botnavigation	3461	\LWR@heavyrulewidth (length)	314
\LWR@caption@begin	6348	\LWR@hidelatexequation	6798
\LWR@caption@end	6358	\LWR@hline	5943
\LWR@clearmidrules	5494	\LWR@hspace	7952
\LWR@closeparagraph	3835	\LWR@htmlblockcomment	3665
\LWR@closeprevious	3481	\LWR@htmlblocktag	3673
\LWR@closetabledat	5080	\LWR@htmlclosecomment	3637
\LWR@cmidrulewidth (length)	314	\LWR@htmlcomment	3658
LWR@colafterspec	301	\LWR@htmldivclass	3698
LWR@colatspec	301	\LWR@htmldivclassend	3701
LWR@colbangspec	301	\LWR@html	3711
LWR@colbarspec	301	\LWR@html	3685
LWR@colbeforespec	301	\LWR@html	3690
\LWR@copyfile	525	\LWR@html	3704
\LWR@createautosec	4267	\LWR@html	3714
LWR@currentautosec (counter)	254	LWR@html	217
\LWR@currentcss	3941	\LWR@html	6119
\LWR@currenttextcolor	7994	\LWR@html	6854
\LWR@descitem	4988	\LWR@html	5717
\LWR@docmidrule	5525	\LWR@html	3637
LWR@doingapar (boolean)	231	\LWR@html	3446
LWR@doingcmidrule (boolean)	293	\LWR@HTML	7135
LWR@doinghline (boolean)	293	\LWR@HTML	7156
LWR@doingstartpars (boolean)	231	\LWR@html	3413
LWR@doingtbrule (boolean)	293	\LWR@html	3610
\LWR@domulticolumn	5681	\LWR@html	3618
\LWR@doubledollar	6720	\LWR@html	3628
LWR@emptyatbang (boolean)	294	\LWR@html	3567
\LWR@endhideamsmath	6877	\LWR@HTML	7747
\LWR@endminipage	7509	\LWR@includegraphicsb	105
\LWR@endofline	7927	\LWR@indexitem	6582, 6586, 6590

LWR@infloatrow (boolean)	499	\LWR@nullfonts	7817
\LWR@instertatbangcols	5072	\LWR@nullifyNoAutoSpacing	5960
LWR@intabularmetadata (boolean)	294	\LWR@openparagraph	3813
LWR@isstartingequation (boolean)	389	\LWR@origcolspec	5200
\LWR@itemizeitem	4971	\LWR@originname	244
LWR@latestautopage (counter)	353	\LWR@originnames	251
LWR@lateximagedepth (counter)	386	\LWR@parseaftercolumn	5261
\LWR@lateximagedepthref	6123	\LWR@parseatcolumn	5207
LWR@lateximagenumber (counter)	386	\LWR@parsebangcolumn	5231
\LWR@lateximagenumberref	6126	\LWR@parsebarcolumn	5270
\LWR@lateximagesfile	487	\LWR@parsebeforecolumn	5252
\LWR@lightrulewidth (length)	314	\LWR@parseDcolumn	5297
LWR@Llpage (counter)	386	\LWR@parsenormalcolumn	5281
\LWR@listitem	4942	\LWR@parsepcolumn	5293
\LWR@loadafter	115	\LWR@parsetablecols	5302
\LWR@loadbefore	126	\LWR@patchlists	5010
\LWR@loadnever	135	LWR@prevFileDepth (counter)	253
\LWR@longtabledatacaptiontag	5728	\LWR@printatbang	5404
\LWR@lookforpackagename	422	\LWR@printbartag	5396
\LWR@LwarpEnd	4621, 8118	\LWR@printmccoldata	5638
\LWR@LwarpStart	4552, 8118	\LWR@printmccoltype	5603
\LWR@maketitlesetup	33, 4726	\LWR@printpendingfootnotes	4048
\LWR@maybeinthisautoid	6315	\LWR@printhetitle	4685
\LWR@maybeowntablerow	5374	\LWR@ProvidesPackageDrop	474
\LWR@maybeprintpendingfootnotes	4058	\LWR@ProvidesPackagePass	462
LWR@midrulecounter (counter)	314	\LWR@pushoneclose	4270
LWR@midrules	313	\LWR@quickfile	484
LWR@minipagedepth (counter)	404	\LWR@remembertag	7194
LWR@minipagefullwidth (boolean)	406	\LWR@requesttoc	4614
LWR@minipageheight (length)	405	\LWR@requirepackagenames	418
\LWR@minipagestartpars	7932	\LWR@restoreorigaccents	740
\LWR@minipagestoppars	7935	\LWR@restoreorigformatting	6618
LWR@minipagethispar (boolean)	406	\LWR@restoreoriglists	5028
LWR@minipagewidth (length)	404	\LWR@restoreorigprintxcolor	7452
\LWR@multicolother	5629	\LWR@rotboxorigin	243
\LWR@multicolparttext	5624	\LWR@rotstyle	91
\LWR@multicolskip	5635	\LWR@rule	7995
\LWR@multirowborder	3	\LWR@scalestyle	94
\LWR@mynexttoken	5042	\LWR@section	4281
\LWR@myshorttoc	6410	\LWR@sectionnumber	4264
\LWR@nameref	6116	\LWR@setexparray	3375
LWR@nestspan (environment)	3577	\LWR@setlatestname	6095
\LWR@newautoidanchor	6318	\LWR@setOSWindows	33
\LWR@newhtmlfile	4172	\LWR@sidetoc	6495
\LWR@newlabel	6153	\LWR@singledollar	6738
\LWR@newref	6203	LWR@skipatbang (boolean)	294
LWR@nextautoid (counter)	352	LWR@skippingmcolrowcell (boolean)	293
LWR@nextautopage (counter)	352	LWR@skippingmrowcell (boolean)	293
LWR@nextequation (counter)	373	LWR@spandepth (counter)	231
\LWR@nohspace	7981	\LWR@splabel	6129

mathjax (boolean)	131
mathjax (option)	72, 75, 132
mathsvg (option)	72, 75, 132
mcaption (package)	528
\mcolrowcell	5933
mdframed (package)	529
\mdseries	7851
memoir	
verse	639
memoir (package)	639
meta tag, HTML	
author	84, 239
description	83, 240
generator	262
Title	239
title	84, 239
viewport	263
metalogo (package)	538
microtype (package)	71, 141, 538
midfloat (package)	539
\MiKTeX	8098
minipage	
framed	88
horizontal space between	422
minipage (environment)	7545
\minipagefullwidth	7537
misplaced \noalign	104, 291
misplaced alignment tab character &	102, 289, 296
missing sections	75, 76
moreverb (package)	539
morewrites (package)	541
mparhack (package)	541
\mrowcell	5930
MS-Windows (program)	80, 130
multicol (package)	541
multicolumn	
with multirow	545
\multicolumnrow	35, 5825
multirow	
with multicolumn	545
\multirow	543
multirow (package)	543
multiline (environment)	6934
multiline* (environment)	6958
N	
\Nameref	6224
\nameref	6215
nameref (package)	546
needspace (package)	546
newclude (package)	108, 546
newfloat	
with trivfloat, algorithmic	634
\newfloat	2, 4
newfloat (package)	144
\newline	7925
\newtheorem	283
newtxmath (package)	100
newunicodechar (package)	71, 547
nextpage (package)	547
nicefrac (package)	99
\nolinebreak	7986
\nolinkurl	6248
nonumonpart (package)	547
\nopagebreak	7990
\nopagecolor	71
nopageno (package)	548
\normalfont	7859
nowidow (package)	548
ntheorem (package)	548
\numberline	6513
numbers	
left margin	126
numindex (option) [tocbibind]	96, 622
O	
OpenOffice (program)	48
options:	
[lwarpmk]:	
htmlglossary	95, 494
printglossary	95, 494
[tocbibind]:	
numindex	96, 622
[tocloft]:	
titles	94
BaseJobname	73, 132
HomeHTMLFilename	73, 75, 133
HTMLFilename	73, 75, 133
IndexLanguage	68, 73, 95, 132, 494, 621
latexmk	73, 75, 133
lwarpmk	73, 132
mathjax	72, 75, 132
mathsvg	72, 75, 132
OSWindows	73, 80, 130, 133
warpHTML	72, 131
warpprint	72, 131
xdyFilename	73, 84, 132
\OSPathSymbol	32
OSWindows (option)	73, 80, 130, 133

overpic (package) 560

P

packages
 required 140
packages:
 a4 434
 a4wide 435
 a5comb 435
 abstract 435
 addlines 438
 adjmulticol 437
 afterpackage 142
 afterpage 438
 algorithmicx 439
 alltt 440
 amsmath 144
 amsthm 440
 anonchap 443
 anysize 443
 appendix 444
 arabicfront 444
 array 339
 atbegshi 445
 authblk 445
 babel 108
 balance 446
 bigdelim 447
 bigstrut 448
 bookmark 448
 booktabs 449
 boxedminipage 449
 boxedminipage2e 449
 breakurl 450
 calc 143
 cancel 450
 caption 144
 caption2 451
 ccaption 452
 changepage 452
 chappg 453
 chnpage 452
 cleveref 401
 cmap 71
 color 453
 comment 134
 crop 453
 cuted 454
 cutwin 454
 dblfnote 455

dcolumn 455
draftwatermark 456
easy-todo 456
ebook 457
ellipsis 457
emptypage 458
endnotes 458
enumerate 459
enumitem 459
environ 144
epigraph 460
eso-pic 461
etoolbox 129
everyhook 142
everypage 461
everyshi 462
expl3 142
extramarks 462
fancybox 463
fancyhdr 467
fancyref 467
fancyvrb 469
figcaps 474
filecontents 143
fix2col 474
fixme 474
fixmetodonotes 476
float 476
floatflt 478
floatpag 479
floatrow 479
flushend 485
fncychap 485
fnpos 486
fontenc 71, 486
fontspec 70, 487
footmisc 487
footnote 488
footnotehyper 489
framed 490
ftnright 492
fullpage 492
fullwidth 493
geometry 141, 493
getttitlestring 142
glossaries 94, 494
graphics 495
graphicx 506
grffile 506
hang 507

hyperref	346, 508	overpic	560
hyperxmp	513	pagenote	561
idxlayout	513	paralist	561
ifoddpaper	514	parskip	562
ifplatform	129	pdfscape	562
indentfirst	514	pdfsync	562
inputenc	71, 515	pfnote	563
keyfloat	515	placeins	563
kvoptions	131	prelim2e	563
layout	516	preview	564
letltxmacro	129	quotchap	564
letterspace	517	ragged2e	565
lettrine	517	realscripts	566
lips	518	refcount	144
listings	519	relsize	567
lmodern	70	romanbar	568
longtable	522	romanbarpagenumber	569
lscap	524	rotating	569
lrcaption	524	rotfloat	569
ltxtable	525	savetrees	570
luatodonotes	525	scalefont	571
lwarp	72	scrxextend	571
lwarp-patch-komascript	433	scrhack	574
makeidx	143	scrlayer	574
marginfit	527	scrlayer-notecolumn	576
marginfix	527	scrlayer-scrpage	576
marginnote	528	section	577
mcaption	528	sectsty	578
mdframed	529	setspace	579
memoir	639	shadow	580
metalogo	538	showidx	581
microtype	71, 141, 538	showkeys	581
midfloat	539	sidecap	581
moreverb	539	sidenotes	582
morewrites	541	siunitx	99, 396, 584
mparhack	541	soul	585
multicol	541	soulpos	587
multitrow	543	soulutf8	588
nameref	546	stabular	588
needspace	546	subfig	589
newclude	108, 546	supertabular	594
newfloat	144	syntonly	596
newtxmath	100	tabls	596
newunicodechar	71, 547	tabularx	597
nextpage	547	tabulary	597
nicefrac	99	textarea	598
nonumonpart	547	textcomp	71, 598
nopageno	548	textpos	602
nowidow	548	theorem	603
ntheorem	548	threeparttable	607

tikz	608	pdfLaTeX (program) [requirement]	51
titleps	609	pdfscape (package)	562
titleref	612	pdfseparate (program) [requirement]	51, 55
titlesec	612	pdfsync (package)	562
titletoc	614	pdftocairo (program) [requirement]	51, 55
titling	616	pdftotext (program) [requirement]	51, 55
tocbasic	621	Perl	56
tocbibind	96, 621, 622	perl (program) [requirement]	56
tocloft	623	pfnote (package)	563
tocstyle	630	\phantomsection	8047
todo	630	picture (environment)	403, 7500
todonotes	631	placeins (package)	563
transparent	633	Plastex (program)	47
trivfloat	634	\popclose	3348
typearea	635	Poppler	51, 55
ulem	635	prelim2e (package)	563
units	99	preview (package)	564
upref	637	\printauthor	270, 4693, 4712
verse	638, 639	\printdate	270, 4704, 4714
wallpaper	640	printglossary (option) [lwarpmk]	95, 494
watermark	640	\printindex	6565
wrapfig	641	\PrintStack	3466
xcolor	398, 642	\printthanks	269, 271
xfrac	651	\printtitle	270, 4677, 4711
xifthen	143	problems	121
xltxtra	653	programs	
xmpincl	654	utility	50
xparse	142	programs:	
xpatch	129	[requirement]:	
xstring	143	LuaLaTeX	51
xtab	654	MathJax	51
zref	144	pdfcrop	51
zwpagelayout	656	pdfLaTeX	51
page		pdfseparate	51, 55
inaccessible	75, 76	pdftocairo	51, 55
\pagebreak	7987	pdftotext	51, 55
\pagecolor	70	perl	56
pagenote (package)	561	XeLaTeX	51
\pagenumbering	3499	Adobe	49
\pageref	6210	AsciiDoc	48
\pagerefPageFor	6209	AsciiDoctor	48
\pagestyle	3491	AsciiDoctor-LaTeX	48
Pandoc (program)	48	Flare	49
\paragraph	4474	FrameMaker	49
paralist (package)	561	GELLMU	47
\parbox	7627	GladTeX	48
\parsemulticolumnalignment	5665	Hevea	47
parskip (package)	562	htlatrix	47
\part	4414	InDesign	49
pdfcrop (program) [requirement]	51		

XeLaTeX (program) [requirement]	51	xmpinl (package)	654
\XeTeX	8084	xparse (package)	142
xfrac (package)	651	xpatch (package)	129
\xfracHTMLfontsize	3	xstring (package)	143
xifthen (package)	143	xtab (package)	654
xindy			
customizing	84		
xindy (program)	84		
xltxtra (package)	653		
		Z	
		zref (package)	144
		zwpagelayout (package)	656