



The **lwarp** package

\LaTeX to HTML

v0.49 — 2018/02/19

© 2016–2018 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

Documents may be produced by pdf \LaTeX , Lua \LaTeX , or Xe \LaTeX . A texlua script removes the need for system utilities such as make and gawk, and also supports xindy and latexmk. Configuration is automatic at the first manual compile.

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

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

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

To update existing projects, see [section 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 and related projects:

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

Please consider helping by joining and/or contributing to the 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 country-specific T_EX users groups: <http://tug.org/usergroups.html>

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

Contents

1	Supporting TeX development	2
	List of Figures	39
	List of Tables	39
2	Updates	40
3	Introduction	50
3.1	Supported packages and features	52
4	Alternatives	56
4.1	Internet class	56
4.2	TeX4ht	56
4.3	Translators	56
4.4	AsciiDoc and AsciiDoctor	57
4.4.1	AsciiDoctor-LaTeX	57
4.5	Pandoc	57
4.6	Word processors	57
4.7	Commercial systems	57
4.8	Comparisons	57
5	Installation	59
5.1	Installing the lwarp package	61
5.2	Installing the lwarpmk utility	62
5.2.1	Using a local copy of lwarpmk	64
5.3	Installing additional utilities	64
6	Tutorial	67
6.1	Starting a new project	67
6.2	Compiling the print version with lwarpmk	71
6.3	Compiling the HTML version with lwarpmk	72
6.4	Generating the SVG images	73
6.5	Using MATHJAX for math	74
6.6	Changing the CSS style	75
6.7	Customizing the HTML output	75
6.8	Using latexmk	76
6.9	Using XeLaTeX or LuaLaTeX	77
6.10	Using a glossary	78
6.11	Cleaning auxiliary files	79
6.12	Cleaning auxiliary and output files	79
6.13	Processing multiple projects in the same directory	79
6.14	Using the make utility	79
6.15	Converting an existing document	80

7	Additional details	81
7.1	Font and UTF-8 support	81
7.1.1	Indexes and UTF-8	82
7.2	lwarp package loading and options	83
7.3	Customizing the HTML output	85
7.3.1	Example HTML file naming	89
7.4	Customizing the CSS	90
7.5	Selecting the operating system	91
7.6	Selecting actions for print or HTML output	92
7.7	Commands to be placed into the warpprint environment	93
7.8	Title page	93
7.9	HTML page meta descriptions	94
7.10	HTML page meta title	95
7.11	HTML page meta author	95
7.12	Modifying xindy index processing	95
8	Special cases and limitations	96
8.1	Formatting	96
8.1.1	Text formatting	96
8.1.2	Horizontal space	96
8.1.3	Text alignment	96
8.1.4	Accents	96
8.1.5	Textcomp	97
8.1.6	Superscripts and other non-math uses of math mode	97
8.1.7	Empty \item followed by a new line of text or a nested list:	97
8.1.8	Filenames and URLs in lists or footnotes	97
8.1.9	resize package	97
8.2	Boxes and minipages	98
8.2.1	Marginpars	98
8.2.2	Save Boxes	98
8.2.3	Minipages	98
8.2.4	Side-by-side minipages	99
8.2.5	Framed minipages and other environments	99
8.2.6	fancybox package	100
8.2.7	mdframed package	102
8.3	Cross-references	102
8.3.1	Page references	102
8.3.2	cleveref and varioref packages	103
8.3.3	Hyperlinks, hyperref, and url	103
8.3.4	Footnotes and page notes	103
8.4	Front and back matter	104
8.4.1	Starred chapters and sections	104
8.4.2	abstract package	104
8.4.3	titling and authblk	105
8.4.4	tocloft package	105
8.4.5	appendix package	105

8.4.6	pagenote package	105
8.4.7	endnotes package	105
8.4.8	glossaries package	105
8.4.9	Index and the tocbibind package	106
8.5	Math	107
8.5.1	Rendering tradeoffs	107
8.5.2	SVG option	108
8.5.3	MATHJAX option	109
8.5.4	ntheorem package	110
8.5.5	siunitx package	110
8.5.6	units and nicefrac packages	111
8.5.7	newtxmath package	111
8.6	Graphics	111
8.6.1	grffile package	112
8.6.2	color package	112
8.6.3	xcolor package	112
8.6.4	epstopdf package	113
8.6.5	overpic package	113
8.7	Tabbing	113
8.8	Tabular	113
8.8.1	longtable package	116
8.8.2	supertabular and xtab packages	117
8.8.3	bigdelim package	117
8.9	Floats	118
8.9.1	float, trivfloat, and/or algorithmicx together	118
8.9.2	caption and subcaption packages	118
8.9.3	subfig package	118
8.9.4	floatrow package	118
8.9.5	keyfloat package	119
8.10	Koma-Script	119
8.11	Memoir	119
8.12	Miscellaneous	120
8.12.1	verse and memoir	120
8.12.2	newclude package	121
8.12.3	babel package	121
8.12.4	todonotes and luatodonotes packages	121
8.12.5	fixme	122
8.12.6	xparse	122
9	EPUB conversion	123
10	Word-processor conversion	125
10.1	Activating word-processor conversion	125
10.2	Additional modifications	127
10.3	Recommendations	128
10.4	Limitations	129

11	Modifying lwarp	131
11.1	Modifying a package for lwarp	131
11.1.1	Adding a package to the lwarp.dtx file	132
11.2	Modifying a class for lwarp	132
11.3	Testing lwarp	133
11.4	Modifying lwarpmk	133
12	Troubleshooting	134
12.1	Using the lwarp.sty package	134
12.1.1	Debug tracing output	136
12.2	Compiling the lwarp.dtx file	137
1	lwarp.sty	137
13	Implementation	137
14	Section depths and HTML headings	138
15	Source Code	140
16	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	141
17	pdf_lat_EX T1 and UTF8 encoding	141
18	Unicode input characters	141
19	Miscellaneous tools	142
20	Early package requirements	142
21	Operating-System portability	143
21.1	Common portability code	143
21.2	Unix, Linux, and Mac OS	143
21.3	MS-WINDOWS	143
22	Package options	144
22.1	Conditional compilation	147
23	Package load order	148
23.1	Tests of package load order	148
23.2	Enforcing package loading after lwarp	150
24	Required packages	155
25	Loading packages	160
26	Additional required packages	163

27	File handles	163
28	Include a file	164
29	Copying a file	165
30	Debugging messages	166
31	HTML-conversion output modifications	167
31.1	User-level controls	167
31.2	Heading adjustments	169
32	Remembering original formatting macros	170
33	Accents	173
34	Configuration Files	175
34.1	project_html.tex	175
34.2	lwarpmk.conf	175
34.3	project.lwarpmkconf	176
34.4	lwarp.css	177
34.5	lwarp_sagebrush.css	204
34.6	lwarp_formal.css	209
34.7	sample_project.css	214
34.8	lwarp.xdy	214
34.9	lwarp_mathjax.txt	215
34.10	lwarpmk option	217
35	Stacks	228
35.1	Assigning depths	229
35.2	Closing actions	229
35.3	Closing depths	230
35.4	Pushing and popping the stack	230
36	Data arrays	232
37	Sanitizing labels and filenames	232
38	HTML entities	233
39	HTML filename generation	234
40	Homepage link	236
41	\LWRPrintStack diagnostic tool	237
42	Closing stack levels	237

43	PDF pages and styles	238
44	HTML tags, spans, divs, elements	240
44.1	Mapping \LaTeX Sections to HTML Sections	240
44.2	Babel-French	240
44.3	HTML tags	241
44.4	Block tags and comments	244
44.5	Div class and element class	245
44.6	Single-line elements	246
44.7	HTML5 semantic elements	246
44.8	High-level block and inline classes	247
44.9	Closing HTML tags	248
45	Paragraph handling	249
46	Paragraph start/stop handling	253
47	Page headers and footers	255
48	CSS	256
49	Title, HTML meta author, HTML meta description	256
50	Footnotes	258
50.1	Regular page footnotes	258
50.2	Minipage footnotes	258
50.3	Titlepage thanks	258
50.4	Regular page footnote implementation	259
50.5	Minipage footnote implementation	261
50.6	Printing pending footnotes	262
51	Marginpars	263
52	Splitting HTML files	264
53	Sectioning	270
53.1	User-level starred section commands	271
53.2	Book class commands	271
53.3	Sectioning support macros	272
53.4	<code>\section</code> and friends	279
54	Starting a new file	281
55	Starting HTML output	284
56	Ending HTML output	287
57	Title page	288

57.1	Setting the title, etc.	289
57.2	\if@titlepage	290
57.3	Changes for \affiliation	291
57.4	Printing the thanks	292
57.5	Printing the title, etc. in HTML	292
57.6	Printing the title, etc. in print form	293
57.7	\maketitle for HTML output	294
57.8	\published and \subtitle	297
58	Abstract	298
59	Quote and verse	299
59.1	Citations and attributions	299
59.2	Quotes, quotations	300
59.3	Verse	300
60	Verbatim and tabbing	301
61	Theorems	304
62	Lists	305
62.1	List environment	306
62.2	Itemize	309
62.3	Enumerate	310
62.4	Description	310
62.5	Patching the lists	311
63	Tabular	313
63.1	Limitations	313
63.2	Token lookahead	316
63.3	Tabular variables	316
63.4	Handling &, @, !, and bar	320
63.4.1	Localizing & catcodes	321
63.4.2	Handling &	322
63.4.3	Filling an unfinished row	323
63.5	Handling \\	325
63.6	Parsing @, >, <, !, bar columns	326
63.7	Parsing 'l', 'c', or 'r' columns	329
63.8	Parsing 'p', 'm', or 'b' columns	329
63.9	Parsing 'D' columns	329
63.10	Parsing the column specifications	330
63.11	colortbl and xparse tabular color support	335
63.12	Starting a new row	337
63.13	Printing vertical bar tags	338
63.14	Printing at or bang tags	338
63.15	Data opening tag	339
63.16	Midrules	341

63.17	Cell colors	346
63.18	Multicolumns	349
63.18.1	Parsing multicolumns	349
63.18.2	Multicolumn factored code	352
63.18.3	Multicolumn	355
63.18.4	Longtable captions	355
63.18.5	Counting HTML tabular columns	358
63.19	Multicolumnrow	359
63.20	Utility macros inside a table	361
63.21	Special-case tabular markers	361
63.22	Checking for a new table cell	362
63.23	<code>\mrowcell</code>	365
63.24	<code>\mcolrowcell</code>	365
63.25	New tabular environment	366
64	Cross-references	372
64.1	Setup	372
64.2	Zref setup	374
64.3	Labels	375
64.4	References	377
64.5	Hyper-references	379
65	Floats	383
65.1	Float captions	384
65.1.1	Caption inside a float environment	386
65.1.2	Caption and LOF linking and tracking	386
66	Table of Contents, LOF, LOT	390
66.1	Reading and printing the TOC	390
66.2	High-level TOC commands	393
66.3	Side TOC	394
66.4	Low-level TOC line formatting	395
67	Index and glossary	398
68	Bibliography presentation	401
69	Restoring original formatting	402
70	Math	405
70.1	Limitations	405
70.1.1	Rendering tradeoffs	405
70.1.2	SVG option	405
70.1.3	MATHJAX option	406
70.2	Inline and display math	407
70.3	MATHJAX support	412
70.4	Equation environment	414

70.5	AMS Math environments	416
70.5.1	Support macros	416
70.5.2	Environment patches	417
71	Lateximages	424
71.1	Description	424
71.2	Support counters and macros	425
71.3	Font size	425
71.4	Sanitizing math expressions for HTML	426
71.5	Equation numbers	427
71.6	HTML <alt> tags	429
71.7	lateximage	429
72	center, flushleft, flushright	433
73	Pre-loaded packages	434
74	Siunitx	435
75	Graphics print-mode modifications	436
75.1	General limitations	436
75.2	Print-mode modifications	437
76	Xcolor boxes	437
77	Cleveref	441
78	Picture	444
79	Boxes and Minipages	445
79.1	Counters and lengths	446
79.2	Footnote handling	446
79.3	Minipage handling	446
79.4	Parbox, mbox, makebox, framebox, fbox, raisebox	450
80	Direct formatting	455
81	Skips, spaces, font sizes	461
82	\phantomsection	469
83	\LaTeX and other logos	469
83.1	HTML logos	469
83.2	Print logos	471
84	\AtBeginDocument, \AtEndDocument	472
85	Koma-script	472

86	Memoir	473
87	Trademarks	474
2	lwarp-a4.sty	475
88	a4	475
3	lwarp-a4wide.sty	475
89	a4wide	475
4	lwarp-a5comb.sty	475
90	a5comb	475
5	lwarp-abstract.sty	475
91	abstract	475
6	lwarp-acro.sty	478
92	acro	478
7	lwarp-acronym.sty	479
93	acronym	479
8	lwarp-adjmulticol.sty	481
94	adjmulticol	481
9	lwarp-addlines.sty	482
95	addlines	482
10	lwarp-afterpage.sty	482
96	afterpage	482

11	lwarp-algorithmicx.sty	482
97	algorithmicx	482
12	lwarp-alltt.sty	483
98	alltt	483
13	lwarp-amsthm.sty	484
99	amsthm	484
14	lwarp-anonchap.sty	488
100	anonchap	488
15	lwarp-anysize.sty	488
101	anysize	488
16	lwarp-appendix.sty	489
102	appendix	489
17	lwarp-arabicfront.sty	489
103	arabicfront	489
18	lwarp-array.sty	490
104	array	490
19	lwarp-atbegshi.sty	490
105	atbegshi	490
20	lwarp-authblk.sty	491
106	authblk	491

21	lwarp-backref.sty	492
107	backref	492
22	lwarp-balance.sty	492
108	balance	492
23	lwarp-bigdelim.sty	492
109	bigdelim	492
24	lwarp-bigstrut.sty	494
110	bigstrut	494
25	lwarp-blowup.sty	494
111	blowup	494
26	lwarp-bookmark.sty	494
112	bookmark	494
27	lwarp-booktabs.sty	495
113	booktabs	495
28	lwarp-boxedminipage.sty	495
114	boxedminipage	495
29	lwarp-boxedminipage2e.sty	496
115	boxedminipage2e	496
30	lwarp-breakurl.sty	496
116	breakurl	496

31	lwarp-cancel.sty	497
117	cancel	497
32	lwarp-caption.sty	498
118	caption	498
33	lwarp-caption2.sty	501
119	caption2	501
34	lwarp-ccaption.sty	501
120	ccaption	501
35	lwarp-changebar.sty	501
121	changebar	501
36	lwarp-changepage.sty	502
122	changepage	502
37	lwarp-chngpage.sty	502
123	chngpage	502
38	lwarp-chappg.sty	503
124	chappg	503
39	lwarp-chapterbib.sty	503
125	chapterbib	503
40	lwarp-cite.sty	503

126	cite	503
41	lwarp-color.sty	504
127	color	504
42	lwarp-colortbl.sty	504
128	colortbl	504
43	lwarp-continue.sty	506
129	continue	506
44	lwarp-crop.sty	506
130	crop	506
45	lwarp-cuted.sty	507
131	cuted	507
46	lwarp-cutwin.sty	507
132	cutwin	507
47	lwarp-dblfnote.sty	508
133	dblfnote	508
48	lwarp-dcolumn.sty	508
134	dcolumn	508
49	lwarp-draftwatermark.sty	509
135	draftwatermark	509

50	lwarp-easy-todo.sty	509
136	easy-todo	509
51	lwarp-ebook.sty	510
137	ebook	510
52	lwarp-ellipsis.sty	510
138	ellipsis	510
53	lwarp-emptypage.sty	511
139	emptypage	511
54	lwarp-endfloat.sty	511
140	endfloat	511
55	lwarp-endheads.sty	511
141	endheads	511
56	lwarp-endnotes.sty	512
142	endnotes	512
57	lwarp-enumerate.sty	513
143	enumerate	513
58	lwarp-enumitem.sty	513
144	enumitem	513
59	lwarp-epigraph.sty	514
145	epigraph	514

60	lwarp-eso-pic.sty	515
146	eso-pic	515
61	lwarp-everypage.sty	516
147	everypage	516
62	lwarp-everyshi.sty	516
148	everyshi	516
63	lwarp-extramarks.sty	516
149	extramarks	516
64	lwarp-fancybox.sty	517
150	fancybox	517
65	lwarp-fancyheadings.sty	521
151	fancyheadings	521
66	lwarp-fancyhdr.sty	521
152	fancyhdr	521
67	lwarp-fancyref.sty	522
153	fancyref	522
68	lwarp-fancyvrb.sty	523
154	fancyvrb	523
69	lwarp-figcaps.sty	528

155	figcaps	528
70	lwarp-figsize.sty	529
156	figsize	529
71	lwarp-fix2col.sty	529
157	fix2col	529
72	lwarp-fixme.sty	530
158	fixme	530
73	lwarp-fixmetodonotes.sty	531
159	fixmetodonotes	531
74	lwarp-flafter.sty	532
160	flafter	532
75	lwarp-float.sty	532
161	float and \newfloat	532
76	lwarp-floatflt.sty	533
162	floatflt	533
77	lwarp-floatpag.sty	534
163	floatpag	534
78	lwarp-floatrow.sty	534
164	floatrow	534

79	lwarp-fltrace.sty	540
165	fltrace	540
80	lwarp-flushend.sty	540
166	flushend	540
81	lwarp-fncychap.sty	541
167	fncychap	541
82	lwarp-fnlineno.sty	542
168	fnlineno	542
83	lwarp-fnpos.sty	542
169	fnpos	542
84	lwarp-fontenc.sty	542
170	fontenc	542
85	lwarp-fontspec.sty	543
171	fontspec	543
86	lwarp-footmisc.sty	543
172	footmisc	543
87	lwarp-footnote.sty	544
173	footnote	544
88	lwarp-footnotehyper.sty	545
174	footnotehyper	545

89	lwarp-footnpag.sty	546
175	footnpag	546
90	lwarp-framed.sty	546
176	framed	546
91	lwarp-ftnright.sty	548
177	ftnright	548
92	lwarp-fullpage.sty	549
178	fullpage	549
93	lwarp-fullwidth.sty	549
179	fullwidth	549
94	lwarp-fwlw.sty	549
180	fwlw	549
95	lwarp-geometry.sty	550
181	geometry	550
96	lwarp-glossaries.sty	550
182	glossaries	550
97	lwarp-graphics.sty	552
183	graphics	552
183.1	Graphics extensions	552
183.2	Length conversions and graphics options	552
183.3	Printing HTML styles	555
183.4	\includegraphics	555

183.5 Boxes	560
98 lwarp-graphicx.sty	563
184 graphicx	563
99 lwarp-grffile.sty	563
185 grffile	563
100 lwarp-hang.sty	563
186 hang	563
101 lwarp-hanging.sty	565
187 hanging	565
102 lwarp-hypcap.sty	566
188 hypcap	566
103 lwarp-hypdestopt.sty	566
189 hypdestopt	566
104 lwarp-hypernat.sty	566
190 hypernat	566
105 lwarp-hyperref.sty	567
191 hyperref	567
106 lwarp-hyperxmp.sty	571
192 hyperxmp	571

107	lwarp-hyphenat.sty	571
193	hyphenat	571
108	lwarp-idxlayout.sty	572
194	idxlayout	572
109	lwarp-ifoddpage.sty	573
195	ifoddpage	573
110	lwarp-indentfirst.sty	573
196	indentfirst	573
111	lwarp-inputenc.sty	574
197	inputenc	574
112	lwarp-keyfloat.sty	574
198	keyfloat	574
113	lwarp-layout.sty	576
199	layout	576
114	lwarp-letterspace.sty	577
200	letterspace	577
115	lwarp-lettrine.sty	577
201	lettrine	577
116	lwarp-lineno.sty	578

202	lineno	578
117	lwarp-lips.sty	581
203	lips	581
118	lwarp-listings.sty	581
204	listings	581
119	lwarp-longtable.sty	585
205	longtable	585
120	lwarp-lscape.sty	587
206	lscape	587
121	lwarp-ltcaption.sty	587
207	ltcaption	587
122	lwarp-ltxtable.sty	587
208	ltxtable	587
123	lwarp-luacolor.sty	588
209	luacolor	588
124	lwarp-luatodonotes.sty	588
210	luatodonotes	588
125	lwarp-marginfit.sty	590
211	marginfit	590

126	lwarp-marginfix.sty	590
212	marginfix	590
127	lwarp-marginnote.sty	591
213	marginnote	591
128	lwarp-mcaption.sty	591
214	mcaption	591
129	lwarp-mdframed.sty	592
215	mdframed	592
215.1	Limitations	592
215.2	Package loading	593
215.3	Patches	593
215.4	Initial setup	594
215.5	Color and length HTML conversion	594
215.6	Environment encapsulation	594
215.7	Mdframed environment	596
215.8	Titles and subtitles	597
215.9	New environments	599
130	lwarp-memhfixc.sty	602
216	memhfixc	602
131	lwarp-metalogo.sty	602
217	metalogo	602
132	lwarp-microtype.sty	603
218	microtype	603
133	lwarp-midfloat.sty	604
219	midfloat	604

134	lwarp-midpage.sty	604
220	midpage	604
135	lwarp-moreverb.sty	604
221	moreverb	604
136	lwarp-morewrites.sty	606
222	morewrites	606
137	lwarp-mparhack.sty	606
223	mparhack	606
138	lwarp-multicol.sty	607
224	multicol	607
139	lwarp-multirow.sty	608
225	multirow	608
225.1	Multirow	608
225.2	Combined multicolumn and multirow	610
140	lwarp-multitoc.sty	611
226	multitoc	611
141	lwarp-nameref.sty	611
227	nameref	611
142	lwarp-natbib.sty	611
228	natbib	611
143	lwarp-needspace.sty	612

229	needspace	612
144	lwarp-newclude.sty	613
230	newclude	613
145	lwarp-newunicodechar.sty	613
231	newunicodechar	613
146	lwarp-nextpage.sty	613
232	nextpage	613
147	lwarp-nonumonpart.sty	614
233	nonumonpart	614
148	lwarp-nopageno.sty	614
234	nopageno	614
149	lwarp-nowidow.sty	614
235	nowidow	614
150	lwarp-ntheorem.sty	615
236	ntheorem	615
236.1	Limitations	615
236.2	Options	615
236.3	Remembering the theorem style	616
236.4	HTML cross-referencing	619
236.5	\newtheoremstyle	619
236.6	Standard styles	620
236.7	Additional objects	621
236.8	Renewed standard configuration	622
236.9	amsthm option	623
236.10	Ending a theorem	626
236.11	\NoEndMark	626

236.12 List-of	626
236.13 Symbols	627
236.14 Cross-referencing	627
151 lwarp-overpic.sty	627
237 overpic	627
152 lwarp-pagegrid.sty	628
238 pagegrid	628
153 lwarp-pagenote.sty	628
239 pagenote	628
154 lwarp-pagesel.sty	628
240 pagesel	628
155 lwarp-paralist.sty	628
241 paralist	628
156 lwarp-parskip.sty	629
242 parskip	629
157 lwarp-pdfscape.sty	629
243 pdfscape	629
158 lwarp-pdfrenderer.sty	630
244 pdfrenderer	630
159 lwarp-pdfsync.sty	630

245	pdfsync	630
160	lwarp-pfnote.sty	630
246	pfnote	630
161	lwarp-placeins.sty	631
247	placeins	631
162	lwarp-prelim2e.sty	631
248	prelim2e	631
163	lwarp-prettyref.sty	631
249	prettyref	631
164	lwarp-preview.sty	632
250	preview	632
165	lwarp-quotchap.sty	632
251	quotchap	632
166	lwarp-ragged2e.sty	633
252	ragged2e	633
167	lwarp-realscripts.sty	634
253	realscripts	634
168	lwarp-relsizes.sty	635
254	relsize	635

169	lwarp-resizegather.sty	636
255	resizegather	636
170	lwarp-romanbar.sty	636
256	romanbar	636
171	lwarp-romanbarpagenumber.sty	637
257	romanbarpagenumber	637
172	lwarp-rotating.sty	637
258	rotating	637
173	lwarp-rotfloat.sty	638
259	rotfloat	638
174	lwarp-savetrees.sty	639
260	savetrees	639
175	lwarp-scalefnt.sty	639
261	scalefnt	639
176	lwarp-scrextend.sty	639
262	scrextend	639
177	lwarp-scrhack.sty	642
263	scrhack	642
178	lwarp-scrlayer.sty	642
264	scrlayer	642

179	lwarp-scrlayer-notecolumn.sty	644
265	scrlayer-notecolumn	644
180	lwarp-scrlayer-scrpage.sty	645
266	scrlayer-scrpage	645
181	lwarp-section.sty	646
267	section	646
182	lwarp-sectionbreak.sty	647
268	sectionbreak	647
183	lwarp-sectsty.sty	647
269	sectsty	647
184	lwarp-setspace.sty	648
270	setspace	648
185	lwarp-shadow.sty	649
271	shadow	649
186	lwarp-showidx.sty	649
272	showidx	649
187	lwarp-showkeys.sty	650
273	showkeys	650
188	lwarp-sidecap.sty	650
274	sidecap	650

189	lwarp-sidenotes.sty	651
275	sidenotes	651
190	lwarp-siunitx.sty	652
276	siunitx	652
191	lwarp-soul.sty	657
277	soul	657
192	lwarp-soulpos.sty	659
278	soulpos	659
193	lwarp-soulutf8.sty	659
279	soulutf8	659
194	lwarp-stabular.sty	660
280	stabular	660
195	lwarp-stfloats.sty	660
281	stfloats	660
196	lwarp-subfig.sty	661
282	subfig	661
197	lwarp-subfigure.sty	666
283	subfigure	666
198	lwarp-supertabular.sty	666
284	supertabular	666

199	lwarp-syntonly.sty	668
285	syntonly	668
200	lwarp-tabls.sty	668
286	tabls	668
201	lwarp-tabularx.sty	669
287	tabularx	669
202	lwarp-tabulary.sty	669
288	tabulary	669
203	lwarp-textarea.sty	670
289	textarea	670
204	lwarp-textcomp.sty	670
290	textcomp	670
290.1	Limitations	671
290.2	Package loading	671
290.3	Remembering original defintions	671
290.4	HTML symbols	672
290.4.1	pdf \TeX symbols	672
290.4.2	X \TeX and Lua \TeX symbols	672
290.5	HTML dicritics	673
290.6	Inside a lateximage	673
205	lwarp-textfit.sty	674
291	textfit	674
206	lwarp-textpos.sty	675
292	textpos	675

207	lwarp-theorem.sty	675
293	theorem	675
293.1	Remembering the theorem style	676
293.2	CSS patches	677
208	lwarp-threeparttable.sty	679
294	threeparttable	679
209	lwarp-tikz.sty	680
295	tikz	680
210	lwarp-titleps.sty	681
296	titleps	681
211	lwarp-titleref.sty	685
297	titleref	685
212	lwarp-titlesec.sty	685
298	titlesec	685
213	lwarp-titletoc.sty	687
299	titletoc	687
214	lwarp-titling.sty	689
300	titling	689
215	lwarp-tocbasic.sty	693
301	tocbasic	693
216	lwarp-tocbibind.sty	694

302	tocbibind	694
217	lwarp-tocloft.sty	696
303	tocloft	696
218	lwarp-tocstyle.sty	703
304	tocstyle	703
219	lwarp-todo.sty	703
305	todo	703
220	lwarp-todonotes.sty	704
306	todonotes	704
221	lwarp-transparent.sty	706
307	transparent	706
222	lwarp-trivfloat.sty	707
308	trivfloat	707
	308.1 Combining \newfloat, \trivfloat, and algorithmicx	707
223	lwarp-turnthepage.sty	708
309	turnthepage	708
224	lwarp-typearea.sty	708
310	typearea	708
225	lwarp-ulem.sty	709
311	ulem	709

226	lwarp-upref.sty	711
312	upref	711
227	lwarp-verse.sty	711
313	verse	711
228	lwarp-vertbars.sty	713
314	vertbars	713
229	lwarp-vmargin.sty	714
315	vmargin	714
230	lwarp-vwcol.sty	714
316	vwcol	714
231	lwarp-wallpaper.sty	717
317	wallpaper	717
232	lwarp-watermark.sty	717
318	watermark	717
233	lwarp-wrapfig.sty	718
319	wrapfig	718
234	lwarp-xcolor.sty	719
320	xcolor	719
320.1	Limitations	719
320.2	Xcolor definitions: location and timing	720
320.3	Package loading	722
320.4	Remembering and restoring original definitions	722
320.5	HTML color style	722

320.6	HTML border	724
320.7	High-level macros	724
320.8	Row colors	728
235	lwarp-xfrac.sty	729
321	xfrac	729
236	lwarp-xltxtra.sty	732
322	xltxtra	732
237	lwarp-xmpinl.sty	733
323	xmpinl	733
238	lwarp-xtab.sty	733
324	xtab	733
239	lwarp-xurl.sty	735
325	xurl	735
240	lwarp-zwpagelayout.sty	735
326	zwpagelayout	735
241	lwarp-patch-komascript.sty	736
327	patch-komascript	736
242	lwarp-patch-memoir.sty	738
328	patch-memoir	738
328.1	Packages	739
328.2	Preliminary setup	740
328.3	Laying out the page	741
328.4	Text and fonts	743

328.5	Titles	744
328.6	Abstracts	744
328.7	Document divisions	744
328.8	Pagination and headers	747
328.9	Paragraphs and lists	749
328.10	Contents lists	749
328.11	Floats and captions	754
328.12	Page notes	759
328.13	Decorative text	761
328.14	Poetry	761
328.15	Boxes, verbatims and files	761
328.16	Cross referencing	762
328.17	Back matter	763
328.18	Miscellaneous	763
328.19	Final patchwork	764

Change History and Index **765**

List of Figures

1	tutorial.tex listing	68
---	--------------------------------	----

List of Tables

1	\LaTeX -HTML generation — lwarp package — Supported features . . .	52
2	Required software programs	60
3	Files created along with the print version	70
4	Package options	83
5	HTML settings	85
6	Section HTML headings for word-processor conversion	129
7	Section depths and HTML headings	138
8	Tabular baseline	330
9	Tabular HTML column conversions	331
10	Cross-referencing data structures	373
11	Float data structures	383
12	AMStm package — CSS styling of theorems and proofs	484
13	Ntheorem package — CSS styling of theorems and proofs	615
14	Theorem package — CSS styling of theorems and proofs	676

2 Updates

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

For a detailed list of changes, see the Change History on page [765](#).

v0.49:

- [tabular](#)
 - Added xcolor `\rowcolors`.
 - Fix: `\noalign` inside a `tabular`.
- [math](#)
 - Fix: `\eqref` in a caption.
- [misc fixes](#)
 - Fix: Incorrect PDF font size changes caused occasional HTML corruption.
 - Fix: `printlen` changes are now grouped for HTML output.
- [packages](#)
 - Added `vwcol`, `vertbars`, `hyphenat`, `lineno`, `fnlineno`, `figsize`, `hypdestopt`, `pagegrid`, `pdfrender`, `luacolor`, `resizegather`.

v0.48:

- [docs](#)
 - Added some documentation regarding converting an existing document. See section [6.15](#).
- [cleveref](#)
 - Updated compatibility for new `cleveref` v0.21.
- [tabular](#)
 - Fix: Ignores optional `tabular` column arguments.
- [minor updates](#)
 - Added `\leftline`, `\centerline`, `\rightline`.
 - Lists have improved font control via `\makelabel`.
 - Print-mode `lateximage` now boxed to the natural width of its multiline contents.
 - `abstract` now allows an optional name, as required by some classes.
- [math](#)
 - Fix: Improved spacing, `\mbox`, and font sizes with `svg math`, `Tikz`.
 - `siunitx`: Improved `svg math`, fraction compatibility, color output.
- [misc. fixes](#)
 - Fix: LOF/LOT links.
 - Fix: Virtual page size grouping caused excessive PDF page breaks.
 - Fix: Parsing similar package names in a single `\usepackage`.
 - Fix: Adapts to classes without `\part`.
 - Fix: `\newline` in `\title` was causing `
` in window title.
 - Fix: `\maketitle` with `\cr`, `\crrcr`, `\noalign`, for `IEEEtran` class.
 - Fix: `xfrac` neutralized `BlockClass` and others.
 - Fix: `todonotes` and `luatodonotes`: Improved `\todototoc`.
- [packages](#)
 - Added `colortbl`, `chapterbib`, `acro`, `acronym`, `hypernat`, `hypcap`, `stfloats`, `vmargin`, `fancyheadings`.
 - `fancyref`: Now directly supported.

v0.47:

math

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

packages

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

v0.46:

name change

misc. fixes

- \PrintStack changed to \LWRPrintStack.
- Fix: Empty lines between tabular rows.
- Fix: Stack unnesting.
- Fix: SVG math and lateximages in numerous situations.
- Fix: Spaces in \usepackage.
- Fix: Now allows MATHJAX inside verse.

v0.45:

docs

- Improved MikTeX install instructions.
- Improved graphics and epstopdf instructions.
- Updates to the [Introduction](#).

memoir

cross-references

- Added memoir, memhfixc. See section [8.11](#).
- Fix: Now allows underscores in labels.
- Fix: _ and \<blank> in section/file names.

math

bibliography

- Fix: Now allows MATHJAX inside tabbing.
- Fix: Bibliography \em names.
- Added cite, natbib, backref. (Also works as-is with biblatex.)

misc. fixes

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

packages

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

v0.44:

- koma-script**
 - Added koma-script classes (except 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, ifoddpag.
 - Added preview, atbegshi, watermark.
 - Improved tocloft \newlistof and \newlistentry.



v0.43:

- footnotes**
 - Docs: Reorganized HTML customization, added an HTML settings table. See section 7.3.
 - Added FootnoteDepth to control the placement of pending footnotes before section breaks. By default, pending footnotes are printed before each \subparagraph or higher.
- sectioning**
 - Fix: Expansion in section name.
- tabular**
 - Fix: Ignore spaces in tabular column specification.
 - Fix: Tabular rules at bottom or when finishing incomplete rows.
 - Fix: \multicolumn at/bang/before/after specifications, trim, and vertical rules.
 - Fix: supertabular and xtab column misalignment.
- math**
 - Fix: equation*.
 - Fix: svg math in a section name.
 - Fix: \ref and \eqref in svg math.
- packages**
 - Added todonotes and luatodonotes (but only disabled).
 - Added breakurl.
 - hyperref: Fix: Several macros were made robust, \Gauge added.

v0.42:

Support \TeX !

word-processor conversion

-  name change
-  name change

- Added \TeX development support page, [Supporting \$\text{\TeX}\$ development](#).
- 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.

tabbing

overpic

math

- Add: tabbing environment.
- Add: overpic package. See section 237.
- Fix: Text copy/paste of \mathcal{MS} 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

-  new syntax

- Added tabular vertical rules, subject to some limitations. See the rules section of section 8.8.
- Improved booktabs: Width and trim are honored.
- Added `\mcolrowcell` for empty cells inside a `\multicolumnrow`. [Use `\mcolrowcell` instead of `\mrowcell` for two-dimensional cells created by `\multicolumnrow`](#). Continue to use `\mrowcell` for empty cells in a `\multirow`. See section 225.2 on section 225.2.
- 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

- `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.



bigdelim
symbols
fixes



margins
columns
footnotes
tabular
sectioning

- grffile is now directly supported instead of emulated.
- Fix for bigdelim, and improved documentation. See section 109.
- Improved \LaTeX and textcomp symbols.
- Fix for \LaTeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for `xltxtra` with \LaTeX .
- Fixes for `tocbibind` with `\simplechapter`, etc.
- Fixes for `\multicolumnrow` and `\nullfonts` with older versions of `multirow` and `xparse`.
- Added `\underline`.
- Added `adjmulticol`.
- Added `cuted`, `midfloat`.
- Added `pfnote`, `fnpos`, `dblfnote`.
- Added `stabular`, `tabls`.
- Added `sectsty`, `anonchap`, `quotchap`.

v0.39:

title pages

 `\published and`
`\subtitle`
 `load order`
tabular
multi column/row cell

 macros inside tabular
 tabular defined inside
another environment
tabular
margins
page layout

- 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 57.8. `authblk` is added, and should be loaded before titling. See section 57.
- `\multirow` now supports the new optional `vpos` argument.
- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 225.2.
- Tabular special cases:
 - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 8.8.
 - Added `\ResumeTabular` for use when a tabular environment is defined inside another environment. See section 8.8.
- Added `supertabular`, `xtab`, `bigstrut`, `bigdelim`.
- Added `fullwidth`.
- Added `addlines`, `anysize`, `a4`, `a4wide`, `a5comb`, `textarea`, `zwpgelayout`, `typearea`, `ebook`.

v0.38:

forced single-pass compile

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

starred sections	<ul style="list-style-type: none"> Added <code>\ForceHTMLPage</code> and <code>\ForceHTMLTOC</code> to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 8.4.1.
updated tutorial	<ul style="list-style-type: none"> Modified the tutorial to use the new <code>\ForceHTMLPage</code> and <code>\ForceHTMLTOC</code> macros.
packages	<ul style="list-style-type: none"> Added appendix, tocbibind, fncychap, fix2col.
font size	<ul style="list-style-type: none"> Added relsize, scalefnt. Added realscripts, metalogo, xltextra. Added grffile, romanbar.
page numbering	<ul style="list-style-type: none"> Added arabicfront, chappg, nonumonpart, nopageno, romanbarpagenumber.
front & back matter	<ul style="list-style-type: none"> Docs: Improved description of the use of front/back matter. See section 8.4. Fix: color requests xcolor. Fix: <code>\part</code> for article class.
v0.37:	
<code>\include</code> for HTML	<ul style="list-style-type: none"> <code>\include</code> now maintains independent .aux files for HTML versions.
latexmk	<ul style="list-style-type: none"> comment, used by lwarp, now maintains independent cut files for print and HTML versions, helping latexmk to better know whether to recompile.
accents and symbols	<ul style="list-style-type: none"> Improved support for \TeX accents, textcomp, siunitx symbols.
babel-french	<ul style="list-style-type: none"> Improved babel-french handling for load order and \sim tilde.
v0.36:	
<ul style="list-style-type: none"> Recorganized the documentation section regarding special cases and limitations. (Section 8) Improved source formatting. 	
boxes and frames	<ul style="list-style-type: none"> <code>\fbox</code> and related now use <code>\fboxsep</code> and <code>\fboxrule</code>. <code>\makebox</code> and <code>\framebox</code> now use width and position. <code>\fcolorbox</code> and related now work inside a <code>lateximage</code>.
babel-french	<ul style="list-style-type: none"> babel-french: Improvements for French variants, load order, footnotes, ellipses.
footnotes	<ul style="list-style-type: none"> Improved footnote numbering. <code>lateximage</code> footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with <code>MATHJAX</code>.
siunitx	<ul style="list-style-type: none"> Improved siunitx units. Fix for filenames while using <code>MATHJAX</code>. Fix for <code>\rule</code> when xcolor is not loaded. Added transparent, upref.

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

v0.34:

⚠ Optional arguments

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

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

instead of:

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

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

⚠ Optional arguments

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

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

spans with minipages

framing minipages

lateximage, svg math, tabular

eqnarray

verbatim packages

framing packages

list packages

babel-french

- 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`.
- Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.
- Improved font control for `lateximage`s and `svg math`.
- Added the `eqnarray` environments.
- `fancyvrb` is no longer required (preloaded), but is still supported.
- Added `verbatim` and `moreverb`.
- Added `fancybox`, `boxedminipage2e` and `shadow`.
- `enumitem` is no longer required, but is still supported.
- Added `enumerate` and `paralist`.
- `titleps` is no longer required, but is still supported.
- Added `crop`.
- Added `rotfloat`, `marginfit`, and several minor packages; see the change log.
- Adds fixed-width HTML spaces around punctuation when using `babel-french`. `LuaTeX` does not yet use the extra punctuation spacing.

v0.33:

- Tabular @ and ! columns now have their own HTML columns.
- & catcode changes are localized, perhaps causing errors about the tab alignment character &, so any definitions of macros or environments which themselves contain tabular and & must be enclosed within `\StartDefiningTabulars` and `\EndDefiningTabulars`. See section 63.4.1. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the & character inside the definition. This may include the use inside conditional expressions.
- Several math environments were incorrectly placed inline. Also, for `amsmath` with `svg` math, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.

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

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

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

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

v0.30: **lwarp-newproject**

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

To modify existing documents, remove from the document source:

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

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

 **HTML setup changes.**

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

Old Macro	New Package Option
<code>\HomeHTMLFileName</code>	<code>HomeHTMLFilename</code>
<code>\HTMLFileName</code>	<code>HTMLFilename</code>
<code>\useLatexmk</code>	<code>latexmk</code>
<code>\warpOSWindows</code>	<code>OSWindows</code>

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

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

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

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

- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
 1. Make the changes shown above.
 2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
 3. The old file `lwarp_html.xdy` may be deleted.
- The new lwarp package option `xdyFilename` may be used to tell lwarpmk to use a custom `.xdy` file instead of `lwarp.xdy`. See section 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, luatex.
 - 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 HTML5 interpretation, with only minor intervention on the user's part. No attempt has been made to force \LaTeX to provide for every HTML-related possibility, and HTML cannot exactly render every possible \LaTeX concept. Where compromise is necessary, it is desirable to allow the print output to remain typographically rich, and compromise only in the HTML conversion.

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

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

As a package running directly in \LaTeX , lwarp has some advantages over other methods of HTML conversion. A deeper level of \TeX understanding is of course possible, as \TeX itself is still in use. Lua expressions are still available with Lua \TeX . Entire categories of \LaTeX packages work as-is when used with lwarp: definitions, file handling, utilities, internal data structures and calculations, and anything generating plain-text output. Blocks of PDF output may be automatically converted to SVG images while using the same font and spacing as the original print document, directly supporting Tikz and picture. Numerous packages are easily adapted for HTML versions, either by loading and patching the originals, or by creating nullified or emulated replacements, and all without resorting to external programming. As a result, several hundred packages have already been adapted (table 1), and an uncounted number more work as-is.

Packages have been selected according to several criteria: importance, popularity lists, recent CTAN updates, CTAN topics, mention in other packages, support by other HTML conversion methods, and from sample documents taken from public archives¹.

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

pdf \LaTeX , x \LaTeX , or lua \LaTeX may be used, allowing lwarp to process the usual image formats. While generating HTML output, SVG files are used in place of PDF. Other formats such as JPG are used as-is.

¹An amazing number of decades-old packages are still in modern use.

SVG images may be used for math, and are also used for `picture`, and `Tikz` environments. This format has better browser and e-book support than MathML (as of this writing), while still allowing for high-quality display and printing of images (again, subject to potentially bug-ridden² 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 \TeX source for the expression, allowing it to be copy/pasted into other documents.³ Custom \TeX macros may be used as-is in math expressions, since the math is evaluated entirely inside \TeX .

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

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

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

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

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

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

⁴One SVG image file per math expression. A common scientific paper can easily run into several thousand files, depending on how often something like $\$x\$$ is used. In testing one sample document it appears that hashing would only reduce the number in half.

3.1 Supported packages and features

Table 1 lists some of the various \LaTeX features which may be used.

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

Category	Status and supported features.
Engines:	pdf \LaTeX , X \LaTeX , Lua \LaTeX
Classes:	book, report, article, scrbook, scrreprt, scrartcl, memoir.
Koma-script:	scrxextend, scrlayer, scrhack. Others as listed below.
Memoir:	memhfixc
Page layout:	geometry, fancyhdr, titleps, scrlayer-scrpage, tpearea, vmargin, addlines, anysize, a4, a4wide, a5comb, textarea, zwpagelayout, ebook, preview, draftwatermark, watermark, everyshi, atbegshi, continue, fwlw, turnthepage, pagesel, blowup, pagegrid.
Sectioning:	Adds FileDepth for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation are removed from the filenames. titlesec, fncychap, sectsty, section, anonchap, quotchap, sectionbreak.
Table of contents, figures, tables:	Supported, with hyperlinks. tocbibind, titletoc, tocloft, tocbasic, and tocstyle, multitoc.
Title page:	\maketitle, titlepage, titling, authblk.
Front & back matter:	abstract, appendix.
Indexing:	texindy is used, with hyperlinks. idxlayout.
Glossary:	glossaries and xindy are used.
Bibliography:	cite, natbib, biblatex, chapterbib, backref, hypernat.
Cross-references:	hyperref, cleveref, varioref, fancyref, prettyref, titleref, url, breakurl, xurl, bookmark, hypdestopt.

lwarp Supported Functions — continued

Category	Status
Languages:	babel. (polyglossia is untested.)
Margin notes:	marginfit, marginfix, scrlayer-notecolumn.
Footnotes:	Adds FootnoteDepth to print footnotes at section breaks. footnote, footmisc, marginnote, sidenote, pagenote, endnotes, endheads, footnpag, nccfoots.
Math:	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:	Native \LaTeX theorems, theorem, amsthm, ntheorem.
Additional math:	delarray, bm, math fonts via SVG images, resizegather.
Units and fractions:	siunitx, xfrac, nicefrac, units
Floats:	Appear where declared. float, rotfloat, newfloat, caption and subcaption, subfig, subfigure, capt-of, placeins, trivfloat, floatrow, subfloat, keyfloat, wrapfig, cutwin, floatflt, flafter, fltrace, endfloat, hypcap, stfloats.
Tabular:	tabular environment, array, tabularx, tabulary, threeparttable, multirow, longtable, supertabular, xtab, ltxtable, booktabs, colortbl.
Graphics:	graphics and graphicx. \includegraphics supports width, height, origin, angle, and scale tags, and adds class. References to PDF files are changed to SVG, other image types are accepted as well. \rotatebox and \scalebox are supported as well as HTML can handle. rotating is emulated but all objects are unrotated. picture and tikz are converted to an SVG image. grffile, overpic, figsize.

lwarp Supported Functions — continued

Category	Status
xcolor:	Full package color names, any color models, and mixing. <code>\textcolor</code> , <code>\colorbox</code> , <code>\fcolorbox</code> . Enhanced for HTML compatibility.
Lists:	Standard \LaTeX environments, enumitem, enumerate, paralist, hang.
Environments:	Standard \LaTeX environments.
minipage:	Some HTML5-imposed limitations. Nested minipages are supported.
Quotations:	verse, csquotes, epigraph
Verbatim:	verbatim, moreverb, fancyvrb (except for verbatim footnotes), shortvrb.
Frames:	framed, fancybox, mdframed, boxedminipage2e, shadow, vertbars.
Multi-columns:	multicol, adjmulticol, vwcol.
Margins:	midpage, hanging, fullwidth.
Line numbering:	lineno, fmlineno.
Acronyms:	acro, acronym.
Todo notes:	todo, todonotes, easy-todo, fixmetodonotes, fixme, changebar.
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are only supported in a few cases. <code>lettrine</code> , <code>ulem</code> , <code>soul</code> , <code>soulutf8</code> , <code>soulpos</code> , <code>cancel</code> , <code>resize</code> , <code>scalefnt</code> , <code>textfit</code> , <code>realscripts</code> , <code>hyphenat</code> , <code>pdfrender</code> , <code>luacolor</code> .
Ordinals:	<code>nth</code> , <code>fntcount</code> , <code>engord</code> .
Text ligatures:	Ligatures for symbols are supported. Ligatures for f, q, t are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</code> , <code>\hspace</code> .

lwarp Supported Functions — continued

Category	Status
Rules:	<code>\rule</code> with width, height, raise, text color.
HTML reserved characters:	<code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
Fonts:	Used as-is. Appear in svg math expressions or embedded image environments.
Symbols:	Native \TeX diacriticals, <code>textcomp</code> , <code>textgreek</code> .
Working as-is:	Various utility, calculation, file, and text-only packages, such as <code>calc</code> , <code>fileerr</code> , <code>somedefs</code> , <code>trace</code> , <code>xspace</code> .

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 `htlatex` This system uses native \LaTeX processing to produce a DVI file containing special commands, and then uses additional post-processing for the HTML conversion by way of numerous configuration files. In 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` **LaTeXML**: <http://dlmf.nist.gov/LaTeXML/>

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

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

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

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

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

4.4 AsciiDoc and AsciiDoctor

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

Prog AsciiDoc **AsciiDoctor:** <http://asciidoctor.org/> (More active.)
 Prog AsciiDoctor **AsciiDoc:** <http://asciidoc.org/> (The original project.)

4.4.1 AsciiDoctor-LaTeX

The AsciiDoctor-LaTeX project is 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 recent developments in Microsoft® Word® and LibreOffice™ Writer.
 Prog LibreOffice
 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 \LaTeX provides many more

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

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

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

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

5 Installation

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

Detailed installation instructions follow.

Table 2: Required software programs

Provided by your \LaTeX distribution:

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

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

The lwarp package: This package.

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

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

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

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

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

POPPLER PDF utilities:

pdftotext: Used to convert PDF to text.

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

pdftocairo: Used to convert images to SVG.

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

From POPPLER: poppler.freedesktop.org.

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

```
Enter ⇒ brew install poppler
```

For WINDOWS, see:

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

Perl:

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

strawberryperl.com (recommended), perl.org

Automatically downloaded from the internet as required:

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

5.1 Installing the lwarp package

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

Pre-installed: Try entering into a command line:

```
Enter ⇒ kpsewhich lwarp.sty
```

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

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

```
Enter ⇒ tlmgr install lwarp
```

MiK_TE_X: If using MiK_TE_X:

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

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

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

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

T_EX Live:

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

MiK_TE_X:

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

This should be something like:

```
/usr/local/texlive/texmf-local/
```

4. Unpack the archive in the TDS local directory.
5. Renew the cache:

```
Enter ⇒ mktexlsr
```

— or —

```
Enter ⇒ texhash
```

Or, for Windows MiK_TE_X, 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 MiK_T_EX, start the program called MiK_T_EX 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 `rds` 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 60.

```
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 installed in TexLive with tlmgr, installed by MikTeX, provided by your operating system's package manager, or downloaded from the CTAN archive:

```

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

```

```

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 table 2.
2. Download and extract the POPPLER utilities pdftotext, pdfseparate, and pdfseparate to a directory, such as Poppler.
3. In the Start window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose "Edit the system environment variables" in the control panel.
5. Choose the "Environment Variables" button.
6. Choose the "Path" variable, then the "Edit" button.
7. Choose the "New" button to make an additional entry.
8. Enter the bin directory of the POPPLER utilities, such as:

```
C:\Users\<myname>\Desktop\Poppler\poppler-0.5_x86\poppler-0.5\bin
```

Be sure to include \bin.
9. Click "Ok" when done.

Prog perl **To install Perl to a Windows machine:**

1. Download and install a version of Perl, such as 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.

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

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

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

```

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

\begin{document}

\maketitle                     % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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


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

\end{document}

```

Table 3: Files created along with the print version

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

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

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

Enter \Rightarrow `lwarpmk print`

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

4. Force a recompile to occur.

Enter \Rightarrow `lwarpmk again`

Enter \Rightarrow `lwarpmk print`

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

5. Process the index.⁶⁷

Enter \Rightarrow `lwarpmk printindex`

6. Recompile again to include the index.

Enter \Rightarrow `lwarpmk print`

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

Enter \Rightarrow `lwarpmk print1`

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

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

⁶A `lwarpmk printglossary` command is also available to process a glossary produced with the glossaries package. See section 8.4.8.

⁷Also 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.html`⁸, `Some-math.html`, and the document's index in `_Index.html`.⁹

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:¹⁰¹¹

```
Enter ⇒ lwarpmk htmlindex
```

```
Enter ⇒ lwarpmk html
```

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

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

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

```
Enter ⇒ lwarpmk html1
```

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

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

¹⁰A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 8.4.8.

¹¹Also 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.

 **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.

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_gLaTeX 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_gLaTeX or LuaLaTeX will be used for future runs of lwarpmk.

3. To recompile the document:

```
Enter ⇒ lwarpmk print
```

-and-

```
Enter ⇒ lwarpmk html
```

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

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.

6.15 Converting an existing document

To convert an existing document for use with lwarp:

1. Arrange the document in the following order:
 - (a) Declare the `\documentclass`.
 - (b) Load text fonts.
 - (c) Load `inputenc`, `fontenc`, and/or `fontspec`.
 - (d) Load lwarp.
 - (e) Load remaining packages.
2. Also modify the document:
 - (a) Change `\includegraphics` PDF filenames from `filename.pdf` to `filename` without a suffix. Other image formats may be left unmodified, or may be loaded without a file suffix.
 - (b) Possible changes to tabular environments: `* columns`, `multirow`, `longtable`, `supertabular`, `xtab`, `bigdelim`. See section 8.8.
 - (c) Possible option clashes with `memoir`. See section 8.11.
 - (d) Other changes as per section 8.
3. Create an SVG version of any PDF image.
4. Manually compile print version with `pdflatex`, `lualatex`, or `xelatex`.
5. `lwarpmk print` to finish the print version.
6. `lwarpmk html` to create the HTML version.
7. `lwarpmk limages` to create the svg images of any `svg math`, `lateximage`, `Tikz`, etc.

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

vector fonts 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. `\usepackage{newtxmath}` or other math-related font packages. Many of these load `amsmath`, which must be loaded after `lwarp`, so they must also be loaded after `lwarp`.
5. ... the rest of the preamble and the main document.

7.1.1 Indexes and UTF-8

`lwarp` uses the `xindy` program to process 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. \XeTeX or \LuaTeX 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 xindy language option used for index and glossary generation.
<code>latexmk</code>	Boolean for <code>lwarpmk</code> to use <code>latexmk</code> for compiling documents.
<code>lwarpmk</code>	Generate a local copy of <code>lwarpmk.lua</code> .
<code>xdyFilename</code>	Set a custom filename for xindy.

7.2 lwarp package loading and options

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

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 22.
Opt	HomeHTMLFilename	See section 7.3.
Opt	HTMLFilename	See section 7.3.
Opt	IndexLanguage	If using an index or glossary, see section 22.
	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 22.
Opt	xdyFilename	The default xindy filename is lwarp.xdy. If you wish to use a custom .xdy file for index generation, see section 22.

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 Default: <code>\BaseJobname</code>	HomeHTMLFilename: Filename of the homepage, without the “.html” suffix. Defaults to the <code>\BaseJobname</code> . A common setting is: <code>HomeHTMLFilename=index</code> causing the homepage to be the file <code>index.html</code> . Underscores are allowed in <code>HomeHTMLFilename</code> and <code>HTMLFilename</code> options, but may need to be escaped elsewhere, such as when appearing in a list: <code>\item [\href{file_name.pdf}{text}] \</code> See section 7.3.1 for examples of naming and numbering HTML files.
Opt	HTMLFilename Default: <code><empty></code>	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.
Opt	latexmk Default: <code>false</code>	latexmk: Controls whether lwarp uses latexmk to compile the document. This setting is written to lwarpmk’s configuration files.
Opt	mathsvg Default: <code>true</code>	mathsvg: Selects SVG display for math output. (The default.)
Opt	mathjax Default: <code>false</code>	mathjax: Selects MATHJAX for math output.

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

Ctrl	tocdepth	tocdepth: Sectioning depth of the table of contents. See section 14 for a list of L ^A T _E X stack depths.
Ctrl	SideTOCDepth Default: <code>1</code> sideroc	SideTOCDepth: Sectioning depth of the sideroc. Defaults to 1, causing the sideroc to show sections but not subsections. Each subpage of the website has its own small table of contents on the side (the “sideroc”). Its depth is set by <code>SideTOCDepth</code> . 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 pagess of the website.

`\HTMLLanguage`
Default: `en-US`

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

`\HTMLTitle`
Default: `\thetitle`

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

`\HTMLAuthor`
Default: `\theauthor`

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

`\HTMLDescription`
Default: `<empty>`

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

`\HTMLFirstPageTop`
Default: `<empty>`

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


`\HTMLPageTop`
Default: `<empty>`

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

`\HTMLPageBottom`
Default: `<empty>`

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

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

 `\tableofcontents`
TOC on the homepage!

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

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

Placed in the document wherever necessary:

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

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

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

`\warpHTMLonly` **\warpHTMLonly:** `{\langle contents \rangle}` A macro version of the warpHTML 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`.¹²

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

¹²See `\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`
Default: `lwarp.css`

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

```
...
\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`
File `project.css`
File `sample_project.css`

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

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

/* 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 MS-WINDOWS is not correctly detected, use the <code>lwarp</code> option <code>OSWindows</code> .
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 \TeX print-formatted PDF generation, or to HTML generation.

For most of built-in \TeX 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 57.8.

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

Env `titlepage` The `titlepage` environment may be used to hold a custom title page. The `titlepage`

will be set in a `<div>` class `titlepage`, and `\printtitle`, etc. may be used inside this environment.

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

`\title` `{<title>}`



Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use a `\subtitle` command instead (section 57.8). The title will appear in the document `\maketitle` as a heading `<h1>`. The HTML meta title tag will also have this title, unless `\HTMLTitle` is used to set the meta title to something else instead.

`\author` `{<author>}`



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

`\date` `{<date>}`

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

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

7.11 HTML page meta author

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

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

7.12 Modifying xindy index processing

Prog xindy lwarpmk 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

-  **\backslash bfseries, etc.** \backslash textbf, etc. are supported, but \backslash bfseries, etc. are not yet supported.
-  **HTML special chars** $\&$, $<$, and $>$ have special meanings in HTML. If \backslash \&, \backslash textless, and \backslash 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

- \backslash hspace \backslash 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 \backslash fill is converted to a \backslash qqquad.
- \backslash , \sim and \backslash , are converted to HTML entities.
- \backslash kern \backslash kern and \backslash hskip are entered into the HTML PDF output as-is, then interpreted by pdftotext, and thus usually appear as a single space.
- \backslash hskip

8.1.3 Text alignment


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

8.1.4 Accents

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

8.1.5 Textcomp

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

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

8.1.6 Superscripts and other non-math uses of math mode

Use `x` instead of \textasciix

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

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

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

While creating SVG math for HTML, the original definitions are 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 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.


8.2.2 Save Boxes

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

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 \LaTeX 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`, `\qquad`, `\hspace`, or `\hfill`. The result is similar in print and HTML. Do not use space commands at the start or end of the line.

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.

For inline minipage and lists:

To frame the contents on their own line with improved formatting in HTML: A new command `\fboxBlock` is included, intended to be a direct replacement for `\fbox` for cases where the `\fbox` surrounds a minipage, table, or list. For print output, this behaves as `\fbox`. For HTML output, the contents are placed inside an HTML `<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 \LaTeX 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 display `tabular`,
`minipages`, and lists:

To create a framed minipage in both print and HTML: A new environment `fminipage` is included. For print output, this is identical to `minipage`, except that it is also framed. For HTML output, this forms a `<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 320 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 150 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

Pkg `fancybox`
framed equation example

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

```

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

```

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

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

```

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

```

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

```

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

```

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

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

8.2.7 mdframed package

Pkg **mdframed** 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, underscore, and perhaps the colon (if not French).

 **underscores** `\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.

8.3.1 Page references

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

Ex:

`\ref{sec:name}` on page `\pageref{sec:name}`
in HTML becomes:

“Sec. 1.23 on page (see sec. 1.23)”.

`\pagerefPageFor` may be redefined to “page for”, empty, etc. See section 64.4.

8.3.2 cleveref and varioref packages

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

⚠ **cleveref page numbers**

Ex:

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

See `\cpagerefFor` at section 77 to redefine the message which is printed for page number references.

8.3.3 Hyperlinks, hyperref, and url

Pkg `hyperref` lwarp emulates hyperref, including the creation of active hyperlinks, but does not
Pkg `url` 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.

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

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

`Pkg abstract` 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.

⚠ missing TOC

8.4.3 titling and authblk

Pkg titling

Pkg authblk

package support

⚠ load order

\published and \subtitle

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

If using the `titling` package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 57.8.

8.4.4 tocloft package

Opt tocloft titles

Pkg tocloft

⚠ 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.

8.4.5 appendix package

Pkg appendix

⚠ incorrect TOC link

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

8.4.6 pagenote package

Pkg pagenote

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

8.4.7 endnotes package

Pkg endnotes

table of contents

To place the endnotes in the TOC, use:

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

HTML page

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

```
\ForceHTMLPage
\theendnotes
```

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	<code>IndexLanguage</code>	The <code>lwarp</code> package takes an option <code>IndexLanguage=english</code> to set the language used by <code>xindy</code> . This is passed to <code>xindy</code> using its <code>-L</code> option, and is used for both index and glossary generation.
-----	----------------------------	---

Opt	<code>lwarpmk</code>	<code>printglossary</code>	The <code>lwarpmk</code> package has the commands <code>lwarpmk printglossary</code> and <code>lwarpmk htmlglossary</code> to process the glossaries created by <code>glossaries</code> using <code>xindy</code> .
Opt	<code>lwarpmk</code>	<code>htmlglossary</code>	

8.4.9 Index and the `tocbibind` package

Pkg	<code>makeidx</code>	The <code>lwarp</code> package takes an option <code>IndexLanguage=english</code> to set the language used by <code>xindy</code> . This is passed to <code>xindy</code> using its <code>-L</code> option, and is used for both index and glossary generation.
Pkg	<code>tocbibind</code>	
Opt	<code>IndexLanguage</code>	

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

placement and toc options

An index may be placed inline with other HTML text, or on its own HTML page:

Inline, with a manual TOC entry:

A commonly-used method to introduce an index in a \LaTeX document:

```
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex
```

On its own HTML page, with a manual TOC entry:

```
\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex
```

Inline, with an automatic TOC entry:

Pkg `tocbibind`

The `tocbibind` package may be used to automatically place an entry in the TOC.

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex
```

On its own HTML page, with an automatic TOC entry:

```
\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex
```

Opt `tocbibind` `numindex`
[numbered index section](#)

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

See section 67 for `lwarp`'s core index and glossary code, and section 302 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 \LaTeX into the initial PDF file using the current font¹³, then is captured from the PDF and converted to svg graphics via a number of utility programs. The svg format is a scalable-vector web format, so math may be typeset by \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML alt attribute carries the \LaTeX code which generated the math, allowing copy/paste of the \LaTeX math expression into other documents.

- SVG image font size** The size of the math and text used in the svg image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:
`\renewcommand{\LateximageFontSizeName}{large}`

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

-  **SVG math in \TeX boxes** SVG math does not work inside \TeX boxes, since a `\newpage` is required before and after each image.

¹³See section 321 regarding fonts and fractions.

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

MATHJAX limitations Limitations when using MATHJAX include:

Prog MathJax

chapter numbers

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

<https://groups.google.com/forum/#!topic/mathjax-users/jUteWUcE2bY>

subequations

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

footnotes in math

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

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.

tabbing

- A tabbing environment is emulated using an HTML `<pre>`. While MATHJAX is enabled inside tabbing, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.

⚠ **siunitx inside an equation**

⚠ other macros and packages

custom MathJax macros

- 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`.
- MATHJAX does not automatically support custom \TeX 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

Pkg `ntheorem`

⚠ 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{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`

fractions

Due to `pdftolatex` limitations, fraction output is replaced by symbol output for `per-mode` and `quotient-mode`.

⚠ math mode required

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

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

8.5.6 units and nicefrac packages

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

8.5.7 newtxmath package

Pkg newtxmath The proper load order is:

⚠ loading sequence

```

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

8.6 Graphics

Pkg graphics For `\includegraphics` with `.pdf` files, the user should provide a `.pdf` image file,
 Pkg graphicx and also a `.svg`, `.png`, or `.jpg` version of the same image. **These should be referred
 to without a file extension:**

⚠ .pdf image files
 ⚠ 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.

Prog pdftocairo To convert a PDF image to SVG, use the utility `pdftocairo`:

```
Enter => pdftocairo -svg filename.pdf
```

If a `.pdf` file is 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
```

Pkg epstopdf For `.eps` files, use `epstopdf` to provide a PDF version, and also provide a SVG version as well.

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`.


 **viewports** 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.

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 \TeX , so expect some ugly results for scaling and rotating.

8.6.1 grffile package

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

8.6.2 color package

Pkg `color` `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

Pkg `xcolor` **`\colorboxBlock` and `\fcolorboxBlock`** `\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a ``. 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 `\rowcolors` is supported, except that the optional argument is ignored so far.

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


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

8.6.4 epstopdf package

Pkg `epstopdf` When using `epstopdf` to convert images to PDF, use the `pdftocairo` utility to also provide an SVG version as well. In the document, refer to the image filename without a suffix. The PDF version will be used in print output, and the SVG version will be used for HTML.

8.6.5 overpic package

Pkg `overpic` 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.

 **scaling**

8.7 Tabbing

The `tabbing` environment works, except that `svg math` and `lateximages` do not yet work inside the environment.

8.8 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 `xtable`.

Defining environments:

⚠ misplaced alignment
alignment tab character &

⚠ floatrow

⚠ tabular inside another
environment

- 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
```

This includes before and after defining any macro which used \ttabbox from floatrow.

- 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

⚠ \multirow

vposn

- Multiple paragraphs in one cell of a p, b, m column must have \newline between paragraphs.
- 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 & ...
```

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.

⚠ `\multicolumn &`
`\multirow`

⚠ `skipped cells`

⚠ `empty cells`

`vposn`

⚠ `macro in a table`
`custom macros`

- See section 225.2 for `\multicolumnrow`.
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.

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 & ...
```

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

- 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`

`@ and !`

`\multirow`

⚠ `\newcolumntype`

- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of `@` and `!` is used at each column, and they are used in that order.
- In `\multirow` cells, the print version may have extra instances of `<`, `>`, `@`, and `!` cells on the second and later rows in the `\multirow` which do not appear in the HTML version.
- `\newcolumntype` is ignored; unknown column types are set to 1.

Rules:

`vertical rules`

`width and trim`

`full-width rules`

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

⚠ `\warpprintonly`
 misplaced `\noalign`

- 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.

- 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.

colortbl:

⚠ **row/cell color**

Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

`colortbl` ignores the overhang arguments.

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 \\}
```

8.8.1 longtable package

Pkg `longtable`

⚠

`Longtable \endhead`, `\endfoot`, and `\endlastfoot` rows are not used for HTML, and these rows should be disabled. Use

```
\warpprintonly{row contents}
```

instead of

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

Doing so helps avoid “Misplaced `\noalign.`” when using `\begin{warpprint}`.

Keep the `\endfirsthead` row, which is still relevant to HTML output.

⚠ `\kill` is ignored, place a `\kill` line inside

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

or place it inside `\warpingprintonly`.

⚠ **lateximage** `longtable` is not supported inside a `lateximage`.

8.8.2 supertabular and xtab packages

Pkg `supertabular` For `\tablefirsthead`, etc., enclose them as follows:

Pkg `xtab` **misplaced alignment**
alignment tab character &
`\StartDefiningTabulars`
`\tablefirsthead`
`...`
`\EndDefiningTabulars`

See section 8.8.

⚠ **lateximage** `supertabular` and `xtab` are not supported inside a `lateximage`.

8.8.3 bigdelim package

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

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

```

      a   b
left { c   d
      e   f
      g   h

```

8.9 Floats

8.9.1 float, trivfloat, and/or algorithmicx together

Pkg float
Pkg trivfloat
Pkg algorithmicx

If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 308.1.

⚠ package conflicts

8.9.2 caption and subcaption packages

Pkg caption
Pkg subcaption

To pass options to `caption`, select the options before loading `lwarp`:

⚠ options

```
\documentclass{article}
...
\PassOptionsToPackage{options_list}{caption}
...
\usepackage{lwarp}
...
\usepackage{caption}
```

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

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

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

8.9.3 subfig package

Pkg subfig

⚠ lof/lotdepth

At present, the package options for `lofdepth` and `lotdepth` are not working. These counters must be set separately after the package has been loaded.

horizontal spacing

In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

8.9.4 floatrow package

Pkg floatrow

Use `\StartDefiningTabulars` and `\EndDefiningTabulars` before and after defining macros using `\ttabbox` with a tabular inside. See section 8.8.

⚠ misplaced alignment tab character & 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.

8.9.5 keyfloat package

Pkg `keyfloat`

⚠ `keywrap`

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

8.10 Koma-Script

Cls `komascript`

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

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

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

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

8.11 Memoir

Cls `memoir`

⚠ options clash

While emulating `memoir`, `lwarp` pre-loads a number of packages (section 328.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading `lwarp`:

```

\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}

```

`\verbfootnote` is not supported.

`\newfootnoteseries`, etc. are not supported.

`lwarp` loads `pagenote` to perform `memoir`'s `pagenote` functions, but there are minor differences in `\pagenotesubhead` and related macros.

Poem numbering is not supported.

The `verbatim` environment does not yet support the `memoir` enhancements. It is currently recommended to load and use `fancyvrb` instead.

The `memoir` glossary system is not yet supported by `lwarpmk`. The `glossaries` package may be used instead, but does require the glossary entries be changed from the `memoir` syntax to the `glossaries` syntax.

8.12 Miscellaneous

8.12.1 verse and memoir

<p>Pkg <code>verse</code></p> <p>Cls <code>memoir</code></p> <p><code>\attrib</code></p>	<p>The documentation for the <code>verse</code> and <code>memoir</code> packages suggest defining an <code>\attrib</code> command, which may already exist in current documents, but it will only work for print output. <code>lwarp</code> provides <code>\attribution</code>, which works for both print and HTML output. To combine the two so that <code>\attrib</code> is used for print and <code>\attribution</code> is used for HTML:</p>
--	---

```

\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</p>
--	---

is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

 **spacing** Horizontal spacing relies on `pdftotext`’s ability to discern the layout (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

8.12.2 newclude package

Pkg `newclude` `newclude` modifies `\label` in a non-adaptive way, so `newclude` must be loaded before `lwarp` is loaded:

 **loading**

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


8.12.3 babel package

Pkg `babel`

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

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

punctuation spaces Also when French is used, `lwarp` creates fixed-width space around punctuation by patching `\FBcolonspace`, `\FBthinspace`, `\FBguillspace`, `\FBmedkern`, `\FBthickkern`, `\FBtextellipsis`, and the tilde. If the user’s document also changes these parameters, the user’s changes should be placed inside a `warpprint` environment so that

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

8.12.4 todonotes and luatodonotes packages

Pkg `todonotes` 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.

Pkg `luatodonotes`

8.12.5 fixme

Pkg `fixme` External layouts (`\fxloadlayouts`) are not supported.



external layouts

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

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

8.12.6 xparse

Pkg `xparse` To remove from the log any warnings about redeclaring objects, place the following before `lwarp` is loaded:

```
\usepackage[log-declarations=false]{xparse}
```

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:

	<i>FormatWP</i>
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 \TeX 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 table 6.

10.3 Recommendations

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

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

Table 6: Section HTML headings for word-processor conversion

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

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

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 \LaTeX 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

To quickly find the source for a package in `lwarp.dtx`, search for `*packagename`, such as `*siunitx`.


Likewise, to quickly find the source for a file in `lwarp.dtx`, search for `*filename`, such as `*lwarp.css`.

Purely text-based packages probably will work as-is when generating HTML.

Look to existing code for ideas on how to expand into new code.

An environment may be converted to a `lateximage` then displayed with an image of the resulting \TeX output. See section 78 for an example of the `picture` environment.

To create a custom HTML block or inline CSS class, see section 44.8.

 **\TeX boxes** Any \TeX boxes must be undone, as SVG math or `lateximages` require `\newpage`, which will not work in a \TeX box.

11.1 Modifying a package for lwarp

If a class loads additional packages, it will be required to modify the class for lwarp, since lwarp must be loaded before most other packages.

To work with lwarp, a class must first set up anything which replicates the functions of the basic \TeX classes, load any required fonts, then load lwarp, then finally load and adjust any other required packages.

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

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

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

Each `lwarp-` package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If “Drop”ped, the original print-version package is ignored, and only the `lwarp-` version is used. Use this where the original print version is useless for HTML. If “Pass”ed, the original package is loaded first, with the user-supplied options, then the `lwarp-` version continues loading as well. See section 236 ([ntheorem](#)) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable without changes, there is no need to create a `lwarp-` version.

11.1.1 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 23.1. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`.

11.2 Modifying a class for `lwarp`

If a class loads additional packages, it will be required to modify the class for `lwarp`, since `lwarp` must be loaded before most other packages.

To work with `lwarp`, a class must first set up anything which replicates the functions of the basic \LaTeX classes, load any required fonts, then load `lwarp`, then finally load and adjust any other required packages.

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 8.8.

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.

Options clash: If using memoir, see section 8.11.

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

“Label(s) may have changed. Rerun to get cross-references right.”:

This warning may repeat endlessly if a math expression is used in a caption. Simple math expressions such as $X=1$ may be replaced with

```
\textit{X}\,=\,1
```

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

“Improper \prevdepth”: lateximages and svg math require `\newpage`, which cannot work inside \TeX boxes or `\ensuremath`. Anything using `\newsavebox`, `\newbox`, `lrbox`, `\savebox`, `\hbox`, `\vbox`, `\usebox`, `\sbox`, etc., must be modified to work without box commands.

If you 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:

- Use lwarpmk again followed by lwarpmk html or lwarpmk print to compile the document one more time.
- Labels with special characters may be a problem. It is best to stick with alpha-numeric, hyphen, underscore, and perhaps the colon (if not French).
`\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.
- `cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to “for”.

Ex:

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

in HTML becomes:

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

See `\cpagerefFor` at section 77 to redefine the message which is printed for page number references.

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

Use Xe_{La}TeX or Lua_{La}TeX.

Floats out of sequence:

Mixed “Here” and floating: Floats [H]ere and regular floats may become out of order. `\clearpage` if necessary.

Caption setup: With `\captionsetup` set the positions for the captions above or below to match their use in the source code.

Print document contains HTML tags:

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

Images are appearing in strange places:

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

labels

⚠ underscores

`\nameref`

⚠ empty link

⚠ cleveref page numbers

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 `tbs 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 ~~crude hacks~~ clever tricks. Reference sources are given for many of the solutions, and a quick internet search will provide additional possibilities.

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

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

Packages were emulated where their primary purpose was visual formatting which is not 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.

Table 7: Section depths and HTML headings

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

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

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.

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

16 Detecting the T_EX Engine — pdf_latex, lua_latex, x_e-_latex

```

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

```

17 pdf_lat_EX T1 and UTF8 encoding

When using pdf_lat_EX, lwarp required T1 and UTF8 encoding.

X_lat_EX and Lua_lat_EX 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_T_EX, preserve upright quotes in verbatim text:

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

19 Miscellaneous tools

`\LWR@providelength` $\{\langle lengthname \rangle\}$ Provides the length if it isn't defined yet.

Used to provide source compatibility for lengths which will be ignored, but might or might not be already provided by other packages.

```
26 \newcommand*\LWR@providelength[1]{%
27   \ifdeflength{#1}{\newlength{#1}}%
28 }
```

Prints a length in the given units, without printing the unit itself.

`\LWR@convertto` $\{\langle dest unit \rangle\} \{\langle length \rangle\}$

```
29 \newcommand*\LWR@convertto[2]{\strip@pt\dimexpr #2*65536/\number\dimexpr 1#1}
```

20 Early package requirements

Pkg etoolbox Provides `\ifbool` and other functions.

Pkg xpatch Patches macros with optional arguments.

```
30 \RequirePackage{etoolbox}[2011/01/03]% v2.6 for \BeforeBeginEnvironment, etc.
31 \RequirePackage{xpatch}
```

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

```
32 \RequirePackage{ifplatform}% sense op-system platform
```

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

```
33 \RequirePackage{letltxmacro}
```

21 Operating-System portability

Prog	Unix	lwarp tries to detect which operating system is being used. UNIX / MAC OS / LINUX
Prog	Mac OS	is the default (collectively referred to as “UNIX” in the configuration files), and MS-
Prog	Linux	WINDOWS is supported as well.
Prog	MS-Windows	If MS-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
Opt	OSWindows	modified, the boolean usingOSWindows is set true. This boolean may be tested by the user for later use.

21.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

```
34 \newbool{usingOSWindows}
35 \boolfalse{usingOSWindows}
```

21.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

21.3 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.

```
37 \newcommand*{\LWR@setOSWindows}
38 {
39 \booltrue{usingOSWindows}
40 \renewcommand*{\OSPathSymbol}{\@backslashchar}
41 }
```

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

```
42 \ifwindows
43 \LWR@setOSWindows
44 \fi
```

22 Package options

Pkg kvoptions Allows key/value package options.

```
45 \RequirePackage{kvoptions}
46 \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:

```
47 \newbool{warpingprint}
48 \newbool{warpingHTML}
49 \newbool{mathjax}
```

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

```
50 \booltrue{warpingprint}%
51 \boolfalse{warpingHTML}%
52 \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.

```
53 \DeclareVoidOption{warpprint}{%
54 \PackageInfo{lwarp}{Using option 'warpprint'}
55 \booltrue{warpingprint}%
56 \boolfalse{warpingHTML}%
57 }
```

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.

```
58 \DeclareVoidOption{warpHTML}{%}
```

```

59 \PackageInfo{lwarp}{Using option 'warpHTML'}%
60 \booltrue{warpingHTML}%
61 \boolfalse{warpingprint}%
62 }

```

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.

```

63 \DeclareVoidOption{mathsvg}{%
64 \PackageInfo{lwarp}{Using option 'mathsvg'}
65 \boolfalse{mathjax}%
66 }

```

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

```

67 \DeclareVoidOption{mathjax}{%
68 \PackageInfo{lwarp}{Using option 'mathjax'}
69 \booltrue{mathjax}%
70 }

```

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.

```

71 \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.

```

72 \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 "}")
```

```

73 \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`:

```

74 \newbool{LWR@creatinglwarpmk}
75 \boolfalse{LWR@creatinglwarpmk}
76
77 \DeclareVoidOption{lwarpmk}{
78 \PackageInfo{lwarp}{Using option 'lwarpmk'}}
79 \booltrue{LWR@creatinglwarpmk}
80 }

```

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.

```

81 \DeclareVoidOption{OSWindows}{
82 \PackageInfo{lwarp}{Using option 'OSWindows'}}
83 \LWR@setOSWindows
84 }

```

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: \lwarp

```

85 \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>

```

86 \DeclareStringOption[] {HTMLFilename}

```

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

```

87 \DeclareBoolOption[false] {latexmk}

```

[Execute options](#) Execute the package options, with the defaults which have been set just above:

```

88 \ProcessKeyvalOptions*\relax

```

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

```

89 \providecommand*{\BaseJobname}{\LWR@BaseJobname}

```

Defaults unless already over-ridden by the user:

```

90 \ifcempty{\LWR@HomeHTMLFilename}{
91 \newcommand*{\HomeHTMLFilename}{\BaseJobname}

```

```

92 }{
93 \csedef{HomeHTMLFilename}{\LWR@HomeHTMLFilename}
94 }
95
96 \csedef{HTMLFilename}{\LWR@HTMLFilename}

```

22.1 Conditional compilation

`\warpprintonly` `{\<contents>}`

Only process the contents if producing printed output.

```

97 \newcommand{\warpprintonly}[1]{\ifbool{warpingprint}{#1}{}}

```

`\warpHTMLonly` `{\<contents>}`

Only process the contents if producing HTML output.

```

98 \newcommand{\warpHTMLonly}[1]{\ifbool{warpingHTML}{#1}{}}

```

`Pkg` `comment` Provides conditional code blocks.

```

99 \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.

```

100 \ifbool{warpingHTML}{
101 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
102 }{}
103
104 \ifbool{warpingprint}{
105 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
106 }{}

107 \excludecomment{testing}

```

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

```

108 \includecomment{warpall}

```

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

`Env` `warpHTML`

For HTML output:

```

109 \ifbool{warpingHTML}{%
110 \includecomment{warpHTML}
111 }
112 {\excludecomment{warpHTML}}%

113 \ifbool{warpingprint}
114 {\includecomment{warpprint}}
115 {\excludecomment{warpprint}}

```

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

```

116 \ifbool{LWR@creatinglwarpmk}
117 {\includecomment{LWR@createlwarpmk}}
118 {\excludecomment{LWR@createlwarpmk}}

```

23 Package load order

Several packages should only be loaded before lwarp, and most others should only be loaded after.

Packages which should only be loaded before lwarp have their own

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

which use `\LWR@loadbefore` to trigger an error if they are loaded after lwarp. Examples include fontspec, inputenc, fontenc, and newunicodechar.

Most packages should be loaded after lwarp. This is enforced by a large number of `\LWR@loadafter` statements, below.

Some packages are emulated by memoir, and so these are tested by `\LWR@notmemoirloadafter`, which does not cause an error if memoir is used.

23.1 Tests of package load order

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

```

119 \newcommand*{\LWR@loadafter}[1]{%
120 \@ifpackageloaded{#1}
121 {
122 \PackageError{lwarp}

```

```

123 {Package #1, or one which uses #1, must be loaded after lwarp}
124 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
125 Package #1 may also be loaded by something else, which must also be moved
126 after lwarp.}
127 }
128 {}
129 }

```

`\LWR@notmemoirloadafter` `{\langle packagename \rangle}` Error if not memoir class and this package was loaded before lwarp.

memoir emulates many packages, and pretends that they have already been loaded.

```

130 \@ifclassloaded{memoir}
131 {\newcommand*\LWR@notmemoirloadafter}[1]{} }
132 {\LetLtxMacro\LWR@notmemoirloadafter\LWR@loadafter}

```

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

```

133 \newcommand*\LWR@loadbefore}[1]{%
134 \@ifpackageloaded{#1}
135 {}
136 {
137 \PackageError{lwarp}
138 {Package #1 must be loaded before lwarp}
139 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
140 }
141 }

```

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

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

```

142 \newcommand*\LWR@loadnever}[2]{%
143 \PackageError{lwarp}
144 {Package #1 is not supported by lwarp's HTML conversion.
145 Package(s) #2 may be useful instead}
146 {Package #1 might conflict with lwarp in some way,
147 or is superceded by another package.
148 For a possible alternative, see package(s) #2.}
149 }

```

23.2 Enforcing package loading after lwarp

Packages which should only be loaded after lwarp are tested here to trip an error of they have already been loaded.

The following packages must be loaded after lwarp:

```

150 \LWR@loadafter{a4}
151 \LWR@loadafter{a4wide}
152 \LWR@loadafter{a5comb}
153 \LWR@notmemoirloadafter{abstract}
154 \LWR@loadafter{acro}
155 \LWR@loadafter{acronym}
156 \LWR@loadafter{adjmulticol}
157 \LWR@loadafter{addlines}
158 \LWR@loadafter{afterpage}
159 \LWR@loadafter{algorithmicx}
160 \LWR@loadafter{alltt}
161 \LWR@loadafter{amsmath}
162 \LWR@loadafter{amsthm}
163 \LWR@loadafter{anonchap}
164 \LWR@loadafter{anysize}
165 \LWR@notmemoirloadafter{appendix}
166 \LWR@loadafter{arabicfront}
167 \LWR@notmemoirloadafter{array}
168 % \LWR@loadafter{atbegshi}% used by morewrites
169 \LWR@loadafter{authblk}
170 \LWR@loadafter{backref}
171 \LWR@loadafter{balance}
172 \LWR@loadafter{bigdelim}
173 \LWR@loadafter{bigstrut}
174 \LWR@loadafter{blowup}
175 \LWR@loadafter{bookmark}
176 \LWR@notmemoirloadafter{booktabs}
177 \LWR@loadafter{boxedminipage}
178 \LWR@loadafter{boxedminipage2e}
179 \LWR@loadafter{breakurl}
180 \LWR@loadafter{cancel}
181 \LWR@loadafter{caption}
182 \LWR@notmemoirloadafter{ccaption}
183 \LWR@loadafter{changebar}
184 \LWR@notmemoirloadafter{changepage}
185 \LWR@notmemoirloadafter{chnngpage}
186 \LWR@loadafter{chappg}
187 \LWR@loadafter{chapterbib}
188 \LWR@loadafter{cite}
189 \LWR@loadafter{color}
190 \LWR@loadafter{colortbl}
191 \LWR@loadafter{continue}

```

```
192 \LWR@notmemoirloadafter{crop}
193 \LWR@loadafter{cuted}
194 \LWR@loadafter{cutwin}
195 \LWR@loadafter{dblfnote}
196 \LWR@notmemoirloadafter{dcolumn}
197 \LWR@loadafter{draftwatermark}
198 \LWR@loadafter{easy-todo}
199 \LWR@loadafter{ebook}
200 \LWR@loadafter{ellipsis}
201 \LWR@loadafter{emptypage}
202 \LWR@loadafter{endfloat}
203 \LWR@loadafter{endheads}
204 \LWR@loadafter{endnotes}
205 \LWR@notmemoirloadafter{enumerate}
206 \LWR@loadafter{enumitem}
207 \LWR@notmemoirloadafter{epigraph}
208 \LWR@loadafter{eso-pic}
209 \LWR@loadafter{everypage}
210 \LWR@loadafter{everyshi}
211 \LWR@loadafter{extramarks}
212 \LWR@loadafter{fancybox}
213 \LWR@loadafter{fancyhdr}
214 \LWR@loadafter{fancyref}
215 \LWR@loadafter{fancyvrb}
216 \LWR@loadafter{figcaps}
217 \LWR@loadafter{figsize}
218 \LWR@loadafter{fix2col}
219 \LWR@loadafter{fixme}
220 \LWR@loadafter{fixmetodonotes}
221 \LWR@loadafter{flafter}
222 \LWR@loadafter{float}
223 \LWR@loadafter{floatflt}
224 \LWR@loadafter{floatpag}
225 \LWR@loadafter{floatrow}
226 \LWR@loadafter{fltrace}
227 \LWR@loadafter{flushend}
228 \LWR@loadafter{fncychap}
229 \LWR@loadafter{fnlineno}
230 \LWR@loadafter{fnpos}
231 % fontenc must be loaded before lwarp
232 % fontspec must be loaded before lwarp
233 \LWR@loadafter{footmisc}
234 \LWR@loadafter{footnote}
235 \LWR@loadafter{footnotehyper}
236 \LWR@loadafter{footnpag}
237 \LWR@loadafter{framed}
238 \LWR@loadafter{ftnright}
239 \LWR@loadafter{fullpage}
240 \LWR@loadafter{fullwidth}
241 \LWR@loadafter{fwlw}
```

```
242 \LWR@loadafter{geometry}
243 \LWR@loadafter{glossaries}
244 % \LWR@loadafter{graphics}% pre-loaded by xunicode
245 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
246 \LWR@loadafter{grffile}
247 \LWR@loadafter{hang}
248 \LWR@loadafter{hanging}
249 \LWR@loadafter{hypcap}
250 \LWR@loadafter{hypdestopt}
251 \LWR@loadafter{hypernat}
252 \LWR@loadafter{hyperref}
253 \LWR@loadafter{hyperxmp}
254 \LWR@loadafter{hyphenat}
255 \LWR@loadafter{idxlayout}
256 \LWR@loadafter{ifoddpages}
257 \LWR@loadafter{indentfirst}
258 % inputenc must be loaded before lwarp
259 \LWR@loadafter{keyfloat}
260 \LWR@loadafter{layout}
261 \LWR@loadafter{letterspace}
262 \LWR@loadafter{lettrine}
263 \LWR@loadafter{lineno}
264 \LWR@loadafter{lips}
265 \LWR@loadafter{listings}
266 \LWR@loadafter{longtable}
267 \LWR@loadafter{lscap}
268 \LWR@loadafter{ltcaption}
269 \LWR@loadafter{ltxttable}
270 \LWR@loadafter{luacolor}
271 \LWR@loadafter{luatodonotes}
272 \LWR@loadafter{marginfit}
273 \LWR@loadafter{marginfix}
274 \LWR@loadafter{marginnote}
275 \LWR@loadafter{mcaption}
276 \LWR@loadafter{mdframed}
277 \LWR@loadafter{memhfixc}
278 \LWR@loadafter{metalogo}
279 \LWR@loadafter{microtype}
280 \LWR@loadafter{midfloat}
281 \LWR@loadafter{midpage}
282 \LWR@notmemoirloadafter{moreverb}
283 % morewrites must be loaded before lwarp
284 \LWR@notmemoirloadafter{mparhack}
285 % \LWR@loadafter{multicol}% loaded by ltxdoc
286 \LWR@loadafter{multirow}
287 \LWR@loadafter{multitoc}
288 \LWR@loadafter{nameref}
289 \LWR@loadafter{natbib}
290 \LWR@notmemoirloadafter{needspace}
291 % newclude must be loaded before lwarp
```

```
292 \LWR@loadafter{newtxmath}
293 % newunicodechar must be loaded before lwarp
294 \LWR@notmemoirloadafter{nextpage}
295 \LWR@loadafter{nonumonpart}
296 \LWR@loadafter{nopageno}
297 \LWR@loadafter{nowidow}
298 \LWR@loadafter{ntheorem}
299 \LWR@loadafter{overpic}
300 \LWR@loadafter{pagegrid}
301 \LWR@notmemoirloadafter{pagenote}
302 \LWR@loadafter{pagesel}
303 \LWR@loadafter{paralist}
304 \LWR@notmemoirloadafter{parskip}
305 \LWR@loadafter{pdfrender}
306 \LWR@loadafter{pdfscape}
307 \LWR@loadafter{pdfsync}
308 \LWR@loadafter{pfnote}
309 \LWR@loadafter{placeins}
310 \LWR@loadafter{prelim2e}
311 \LWR@loadafter{prettyref}
312 \LWR@loadafter{preview}
313 \LWR@loadafter{quotchap}
314 \LWR@loadafter{ragged2e}
315 \LWR@loadafter{realscripts}
316 \LWR@loadafter{relsize}
317 \LWR@loadafter{resizegather}
318 \LWR@loadafter{romanbar}
319 \LWR@loadafter{romanbarpagenumber}
320 \LWR@loadafter{rotating}
321 \LWR@loadafter{rotfloat}
322 \LWR@loadafter{savetrees}
323 % \LWR@loadafter{scalegnt}% loaded by babel-french
324 \LWR@loadafter{scrextend}
325 \LWR@loadafter{scrhack}
326 \LWR@loadafter{scrlayer}
327 \LWR@loadafter{scrlayer-notecolumn}
328 \LWR@loadafter{scrlayer-scrpage}
329 \LWR@loadafter{section}
330 \LWR@loadafter{sectionbreak}
331 \LWR@loadafter{sectsty}
332 \LWR@notmemoirloadafter{setspace}
333 \LWR@loadafter{shadow}
334 \LWR@notmemoirloadafter{showidx}
335 \LWR@loadafter{showkeys}
336 \LWR@loadafter{sidecap}
337 \LWR@loadafter{sidenotes}
338 \LWR@loadafter{siunitx}
339 \LWR@loadafter{soul}
340 \LWR@loadafter{soulpos}
341 \LWR@loadafter{soulutf8}
```

```
342 \LWR@loadafter{stabular}
343 \LWR@loadafter{stfloats}
344 \LWR@loadafter{subfig}
345 \LWR@loadafter{subfigure}
346 \LWR@loadafter{supertabular}
347 \LWR@loadafter{tabls}
348 \LWR@notmemoirloadafter{tabularx}
349 \LWR@loadafter{tabulary}
350 \LWR@loadafter{textarea}
351 % \LWR@loadafter{textcomp}% maybe before lwarp with font packages
352 \LWR@loadafter{textfit}
353 \LWR@loadafter{textpos}
354 \LWR@loadafter{theorem}
355 \LWR@loadafter{threeparttable}
356 \LWR@loadafter{tikz}
357 \LWR@loadafter{titleps}
358 \LWR@loadafter{titlesec}
359 \LWR@loadafter{titletoc}
360 \LWR@notmemoirloadafter{titling}
361 % \LWR@loadafter{tocbasic}% preloaded by koma-script classes
362 \LWR@notmemoirloadafter{tocbibind}
363 \LWR@notmemoirloadafter{tocloft}
364 \LWR@loadafter{tocstyle}
365 \LWR@loadafter{todo}
366 \LWR@loadafter{todonotes}
367 \LWR@loadafter{transparent}
368 \LWR@loadafter{trivfloat}
369 \LWR@loadafter{turnthepage}

370 % \LWR@loadafter{typearea}% preloaded by koma-script classes
371 \LWR@loadafter{ulem}
372 \LWR@loadafter{upref}
373 \LWR@loadafter{varioref}% no lwarp package provided
374 \LWR@notmemoirloadafter{verse}
375 \LWR@loadafter{vertbars}
376 \LWR@loadafter{vmargin}
377 \LWR@loadafter{vwcol}
378 \LWR@loadafter{wallpaper}
379 \LWR@loadafter{watermark}
380 \LWR@loadafter{wrapfig}
381 \LWR@loadafter{xcolor}
382 \LWR@loadafter{xfrac}
383 \LWR@loadafter{xltxtra}
384 \LWR@loadafter{xmpinl}
385 \LWR@loadafter{xtab}
386 \LWR@loadafter{xurl}
387 \LWR@loadafter{zwpagelayout}
```

24 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: 388 \begin{warppall}

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

Detects X_YTeX and Lua_YTeX:

```
389 \RequirePackage{iftex}
390 \newif\ifxetexorluatex
391 \ifXeTeX
392   \xetexorluatextrue
393 \else
394   \ifLuaTeX
395     \xetexorluatextrue
396   \else
397     \xetexorluatexfalse
398   \fi
399 \fi
```

```
400 \end{warppall}
```

for HTML output: 401 \begin{warpHTML}

```
402 \ifxetexorluatex
403 % ^^A \usepackage[no-math]{fontspec}
```

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

```
404 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
405 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
406 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
407 \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.

```

408 \RequirePackage {microtype}
409
410 \microtypesetup{
411     protrusion=false,
412     expansion=false,
413     tracking=false,
414     kerning=false,
415     spacing=false}
416
417 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
418
419 \fi
420
421 \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:

```

420 \begin{warpHTML}
421 \RequirePackage[paperheight=190in,paperwidth=20in,%
422 left=2in,right=12in,%
423 top=1in,bottom=1in,%
424 ]{geometry}
425 \@twosidefalse
426 \@mparswitchfalse
427 \end{warpHTML}

```

for HTML & PRINT:

```

428 \begin{warppall}

```

Pkg xparse

\TeX 3 command argument parsing

```

429 \RequirePackage{xparse}

```

Pkg afterpackage Used to patch titling to add \AddSubtitlePublished. Provided by scrfile or after-package.

```
430 \@ifundefined{AfterPackage}%
431 {
432 \RequirePackage{afterpackage}
433 }{}
```

```
434 \end{warpall}
```

for HTML output: 435 \begin{warpHTML}

Pkg expl3

ℒ_{EX}3 programming

```
436 \RequirePackage{expl3}
```

Pkg gettitlestring

Used to emulate \nameref.

```
437 \RequirePackage{gettitlestring}
```

Pkg everyhook

everyhook is used to patch paragraph handling.

```
438 \RequirePackage{everyhook}
439 \end{warpHTML}
```

for HTML & PRINT: 440 \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>

```
441 \RequirePackage{filecontents}
442
443 \@ifpackagelater{filecontents}{2011/10/09}%
444 {}
445 {
446 \newwrite\fcwrite
447 \let\LWR@origfilec@ntents\filec@ntents
```

```
448 \def\filecontents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilecontents}
449 }
```

```
450 \end{warpall}
```

for HTML output: 451 \begin{warpHTML}

Pkg xifthen

```
452 \RequirePackage{xifthen}
```

Pkg xstring

```
453 \RequirePackage{xstring}
```

Pkg xstring

```
454 \RequirePackage{verbatim}
```

Pkg makeidx

```
455 \RequirePackage{makeidx}
```

```
456 \makeindex
```

Pkg calc

```
457 \RequirePackage{calc}
```

Pkg refcount

Provides \setcounterref, \setcounterpageref, etc.

```
458 \RequirePackage{refcount}
```

Pkg newfloat

```
459 \RequirePackage{newfloat}
```

```
460 \end{warpHTML}
```

for HTML & PRINT: 461 \begin{warpall}

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

```
462 \RequirePackage{environ}
```

```
463 \end{warpall}
```

for HTML output:

```
464 \begin{warpHTML}
```

Pkg **zref** Used for cross-references.

```
465 \RequirePackage{zref}
```

Pkg **amsmath** Preloaded to avoid options clash and to add patches.

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.

```
466 \PassOptionsToPackage{leqno}{amsmath}
```

```
467 \RequirePackage{amsmath}
```

Patches to allow `\eqref` inside a caption:

```
468 \def\maketag@@@#1{#1}
```

```
469 \def\tagform@#1{\maketag@@@{\ignorespaces#1\unskip}}}
```

Pkg **printlen** Used to convert lengths for image width/height options.

```
470 \RequirePackage{printlen}
```

\LWR@printlength $\langle length \rangle$

Prints a length using a locally-controlled unit and space. Rounding is used unless the length is small.

```
471 \newrobustcmd*\LWR@printlength}[1]{%
```

```
472 \begingroup%
```

```
473 \uselengthunit{PT}%
```

```
474 \renewcommand*\unitspace{}%
```

```
475 \ifdimless{#1}{10pt}{%
```

```
476     \printlength{#1}%
```

```
477 }{%
```

```
478     \rndprintlength{#1}%
```

```
479 }%
```

```
480 \endgroup%
```

```
481 }
```

```
482 \end{warpHTML}
```

for PRINT output:

```
483 \begin{warpprint}
```

Pkg varwidth Used for print-mode lateximage:

```
484 \RequirePackage{varwidth}
```

```
485 \end{warpprint}
```

25 Loading packages

for HTML output: 486 \begin{warpHTML}

Remember the original \RequirePackage:

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

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

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

\LWR@parsedrequirepackagenames Stores the parsed list of required package names after spaces are removed and lwarp- is prepended.

```
489 \newcommand*\LWR@parsedrequirepackagenames{}
```

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

Note that argument 4 is passed directly to \StrBetween.

```
490 \newcommand*\LWR@findword[3][,]{%
491   \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
492 }
```

\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.

Looks index deep into the list of package names, \LWR@requirepackagenames, and builds \LWR@parsedrequirepackagenames which is the modified list of names.

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

Find the index'th package name from the list:

```
494 \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.

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

See if the package name was found:

```
496 \IfStrEq{\LWR@strresulttwo}{}%
497 {%
498 }% no filename
499 {% yes filename
```

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

```
500 \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
501 {% lwarp-* file found
502 \ifdefvoid{\LWR@parsedrequirepackagenames}{%
503 \edef\LWR@parsedrequirepackagenames{lwarp-\LWR@strresulttwo}%
504 }{%
505 \edef\LWR@parsedrequirepackagenames{%
506 \LWR@parsedrequirepackagenames,lwarp-\LWR@strresulttwo%
507 }%
508 }%
509 }%
510 {%

511 \ifdefvoid{\LWR@parsedrequirepackagenames}{%
512 \edef\LWR@parsedrequirepackagenames{\LWR@strresulttwo}%
513 }{%
514 \edef\LWR@parsedrequirepackagenames{%
515 \LWR@parsedrequirepackagenames,\LWR@strresulttwo%
516 }%
517 }%
518 }% no lwarp-* file
519 }% yes filename
520 }
```

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

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

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

Redirect up to nine names:

```
522 \renewcommand*{\LWR@requirepackagenames}{#2}%
```

```

523 \renewcommand*{\LWR@parsedrequirepackagenames}{}%
524 \LWR@lookforpackagename{1}%
525 \LWR@lookforpackagename{2}%
526 \LWR@lookforpackagename{3}%
527 \LWR@lookforpackagename{4}%
528 \LWR@lookforpackagename{5}%
529 \LWR@lookforpackagename{6}%
530 \LWR@lookforpackagename{7}%
531 \LWR@lookforpackagename{8}%
532 \LWR@lookforpackagename{9}%

```

\RequirePackage depending on the options and version:

```

533 \IfValueTF{#1}%
534 {% options given
535   \IfValueTF{#3}% version given?
536   {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}[#3]}%
537   {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}}%
538 }%
539 {% no options given
540   \IfValueTF{#3}% version given?
541   {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}[#3]}%
542   {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}}%
543 }%
544 }
545 \LetLtxMacro{\usepackage}{\RequirePackage}

```

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

Uses the original package, including options.

```

546 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
547 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
548 \IfValueTF{#2}
549 {\ProvidesPackage{lwarp-#1}[#2]}
550 {\ProvidesPackage{lwarp-#1}}
551 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
552 \ProcessOptions\relax
553 \IfValueTF{#2}
554 {\LWR@origRequirePackage{#1}[#2]}
555 {\LWR@origRequirePackage{#1}}
556 }

```

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

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

```

557 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
558 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
559 \IfValueTF{#2}
560 {\ProvidesPackage{lwarp-#1}[#2]}
561 {\ProvidesPackage{lwarp-#1}}

```

Ignore all options.

```

562 \DeclareOption*{}

```

Nullifies then processes the options. Seems to be required when options contain curly braces, which were causing “Missing \begin{document}”.

```

563 % \ProcessOptions\relax% original LaTeX code
564 \let\ds@{}@empty% from the original \ProcessOptions
565 \edef\@curroptions{}% lwarp modification to \ProcessOptions
566 \@processoptions\relax% from the original \ProcessOptions
567 }

```

```

568 \end{warpHTML}

```

26 Additional required packages

for HTML output: 569 \begin{warpHTML}

Pkg caption

```

570 \RequirePackage{caption}%

```

```

571 \end{warpHTML}

```

27 File handles

Defines file handles for writes.

for HTML & PRINT: 572 \begin{warppall}

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

```

573 \newwrite\LWR@quickfile%

```

```
574 \end{warpall}
```

for HTML output: 575 \begin{warpHTML}

\LWR@lateximagesfile For lateximages.txt.

```
576 \newwrite\LWR@lateximagesfile
```

```
577 \end{warpHTML}
```

28 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: 578 \begin{warpHTML}

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

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

```
579 \def\@include#1 {%
580 \immediate\openout\LWR@quickfile #1_html_inc.tex% lwarp
581 \immediate\write\LWR@quickfile{\string\input{#1.tex}}% lwarp
582 \immediate\closeout\LWR@quickfile% lwarp
583 \LWR@origclearpage% \changed
584 \if@files
585 \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
586 \fi
587 \@tempswattrue
588 \if@partsw
589 \@tempswafalse
590 \edef\reserved@a{#1}%
591 \@for\reserved@a:=\@partlist\do
592 {\ifx\reserved@a\reserved@b\@tempswattrue\fi}%
593 \fi
```

```

594 \if@tempswa
595   \let\@auxout\@partaux
596   \if@filesw
597     \immediate\openout\@partaux #1_html_inc.aux % changed
598     \immediate\write\@partaux{\relax}%
599   \fi
600   \@input{#1_html_inc.tex}% changed
601   \LWR@origclearpage% changed
602   \@writeckpt{#1}%
603   \if@filesw
604     \immediate\closeout\@partaux
605   \fi
606 \else
607   \deadcycles\z@
608   \@nameuse{cp#1}%
609 \fi
610 \let\@auxout\@mainaux%
611 }

612 \end{warpHTML}

```

29 Copying a file

for HTML output: 613 \begin{warpHTML}

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

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

```

614 \newwrite\LWR@copyoutfile % open the file to write to
615 \newread\LWR@copyinfile % open the file to read from
616
617 \newcommand*\LWR@copyfile[2]{%
618 \LWR@traceinfo{LWR@copyfile: copying #1 to #2}
619
620 \immediate\openout\LWR@copyoutfile=#2
621 \openin\LWR@copyinfile=#1
622 \begingroup\endlinechar=-1
623 \makeatletter
624
625 \LWR@traceinfo{LWR@copyfile: about to loop}
626
627 \loop\unless\ifeof\LWR@copyinfile
628   \LWR@traceinfo{LWR@copyfile: one line}
629   \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline

```

```

630 % \LWR@fileline\par % print the content into the pdf
631 % print the content:
632 \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
633 \repeat
634 \immediate\closeout\LWR@copyoutfile
635 \LWR@traceinfo{LWR@copyfile: done}
636 \endgroup
637 }

638 \end{warpHTML}

```

30 Debugging messages

for HTML & PRINT: 639 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

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

\tracinglwarp Turns on the debug tracing messages.

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

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

```

642 \newcommand{\LWR@traceinfo}[1]{%
643 \ifbool{LWR@tracinglwarp}%
644 {%
645 \typeout{*** lwarp: #1}%
646 % \PackageInfo{lwarp}{#1 : }%
647 }%
648 {}%
649 }

```

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

Default: false

```

650 \newbool{HTMLDebugComments}
651 \boolfalse{HTMLDebugComments}

```

If \tracinglwarp, show where preamble hooks occur:

```

652 \AfterEndPreamble{
653 \LWR@traceinfo{AfterEndPreamble}

```

```

654 }
655
656 \AtBeginDocument{
657 \LWR@traceinfo{AtBeginDocument}
658 }

659 \end{warpall}

```

31 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: 660 \begin{warpall}

31.1 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

```

661 \newbool{FormatEPUB}
662 \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

```

663 \newbool{FormatWP}
664 \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.¹⁴

```
665 \newbool{WPMarkFloats}
666 \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.

```
667 \newbool{WPMarkMinipages}
668 \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.

```
669 \newbool{WPMarkTOC}
670 \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.

```
671 \newbool{WPMarkLOFT}
672 \boolfalse{WPMarkLOFT}
```

Bool WPMarkMath While formatting for word processors, prints math as \TeX code instead of creating
 Default: false

¹⁴Perhaps some day word processors will have HTML import options for identifying <figure> and <figcaption> tags for figures and tables.

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

```
673 \newbool{WPMarkMath}
674 \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 table 6.

```
675 \newbool{WPTitleHeading}
676 \boolfalse{WPTitleHeading}
```

```
677 \end{warpall}
```

31.2 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: 678 \begin{warpHTML}

```
679 \AtBeginDocument{
680 \ifbool{FormatWP}{
681 \@ifundefined{chapter}{
682 \ifbool{WPTitleHeading}{% part and section starting at h2
683 \renewcommand*{\LWR@tagtitle}{h1}
684 \renewcommand*{\LWR@tagtitleend}{/h1}
685 \renewcommand*{\LWR@tagpart}{h2}
686 \renewcommand*{\LWR@tagpartend}{/h2}
687 \renewcommand*{\LWR@tagsection}{h3}
688 \renewcommand*{\LWR@tagsectionend}{/h3}
689 \renewcommand*{\LWR@tagsubsection}{h4}
690 \renewcommand*{\LWR@tagsubsectionend}{/h4}
691 \renewcommand*{\LWR@tagsubsubsection}{h5}
692 \renewcommand*{\LWR@tagsubsubsectionend}{/h5}
693 \renewcommand*{\LWR@tagparagraph}{h6}
694 \renewcommand*{\LWR@tagparagraphend}{/h6}
695 \renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
696 \renewcommand*{\LWR@tagsubparagraphend}{/span}
697 }% WPTitleHeading
```

```

698 {% not WPTitleHeading, part and section starting at h1
699 \renewcommand*{\LWR@tagtitle}{div class="title"}
700 \renewcommand*{\LWR@tagtitleend}{/div}
701 \renewcommand*{\LWR@tagpart}{h1}
702 \renewcommand*{\LWR@tagpartend}{/h1}
703 \renewcommand*{\LWR@tagsection}{h2}
704 \renewcommand*{\LWR@tagsectionend}{/h2}
705 \renewcommand*{\LWR@tagsubsection}{h3}
706 \renewcommand*{\LWR@tagsubsectionend}{/h3}
707 \renewcommand*{\LWR@tagsubsubsection}{h4}
708 \renewcommand*{\LWR@tagsubsubsectionend}{/h4}
709 \renewcommand*{\LWR@tagparagraph}{h5}
710 \renewcommand*{\LWR@tagparagraphend}{/h5}
711 \renewcommand*{\LWR@tagsubparagraph}{h6}
712 \renewcommand*{\LWR@tagsubparagraphend}{/h6}
713 }% not WPTitleHeading
714 }% chapter undefined
715 {% chapter defined
716 \ifbool{WPTitleHeading}{
717 {% not WPTitleHeading, part and chapter starting at h1
718 \renewcommand*{\LWR@tagtitle}{div class="title"}
719 \renewcommand*{\LWR@tagtitleend}{/div}
720 \renewcommand*{\LWR@tagpart}{h1}
721 \renewcommand*{\LWR@tagpartend}{/h1}
722 \renewcommand*{\LWR@tagchapter}{h2}
723 \renewcommand*{\LWR@tagchapterend}{/h2}
724 \renewcommand*{\LWR@tagsection}{h3}
725 \renewcommand*{\LWR@tagsectionend}{/h3}
726 \renewcommand*{\LWR@tagsubsection}{h4}
727 \renewcommand*{\LWR@tagsubsectionend}{/h4}
728 \renewcommand*{\LWR@tagsubsubsection}{h5}
729 \renewcommand*{\LWR@tagsubsubsectionend}{/h5}
730 \renewcommand*{\LWR@tagparagraph}{h6}
731 \renewcommand*{\LWR@tagparagraphend}{/h6}
732 \renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
733 \renewcommand*{\LWR@tagsubparagraphend}{/span}
734 }% not WPTitleHeading
735 }% chapter defined
736 }{}% FormatWP
737 }% AtBeginDocument

738 \end{warpHTML}

```

32 Remembering original formatting macros

for HTML output: 739 \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.

```
740 \LetLtxMacro\LWR@origmbox\mbox
741 \LetLtxMacro\LWR@origmakebox\makebox
742
743 \let\LWR@origcomma\,
744 \let\LWR@origtilde~
745 \let\LWR@origenskip\enskip
746 \let\LWR@origquad\quad
747 \let\LWR@origqquad\qquad
748
749 \let\LWR@orighspace\hspace
750 \let\LWR@orighfill\hfill
751 \let\LWR@orighfil\hfil
752 \let\LWR@origvspace\vspace
753 \let\LWR@orighss\hss
754 \let\LWR@origllap\llap
755 \let\LWR@origrlap\rlap
756 \let\LWR@origfilneg\hfilneg
757
758 \let\LWR@origrule\rule
759 \let\LWR@orighrulefill\hrulefill
760 \let\LWR@origdotfill\dotfill
761
762 \let\LWR@origmedskip\medskip
763 \let\LWR@origbigskip\bigskip
764
765 \let\LWR@origtextellipsis\textellipsis
766
767 \let\LWR@orignormalsize\normalsize
768 \let\LWR@origsmall\small
769 \let\LWR@origfootnotesize\footnotesize
770 \let\LWR@origscriptsize\scriptsize
771 \let\LWR@origtiny\tiny
772 \let\LWR@origlarge\large
773 \let\LWR@origLarge\Large
774 \let\LWR@origLARGE\LARGE
775 \let\LWR@orighuge\huge
776 \let\LWR@origHuge\Huge
777
778 \LetLtxMacro{\LWR@origtextrm}{\textrm}
779 \LetLtxMacro{\LWR@origtextsf}{\textsf}
780 \LetLtxMacro{\LWR@origtexttt}{\texttt}
781 \LetLtxMacro{\LWR@origtextnormal}{\textnormal}
782 \LetLtxMacro{\LWR@origtextbf}{\textbf}
783 \LetLtxMacro{\LWR@origtextmd}{\textmd}
784 \LetLtxMacro{\LWR@origtextit}{\textit}
```

```
785 \LetLtxMacro{\LWR@origtextsl}{\textsl}
786 \LetLtxMacro{\LWR@origtextsc}{\textsc}
787 \LetLtxMacro{\LWR@origtextup}{\textup}
788 \LetLtxMacro{\LWR@origemph}{\emph}
789
790 \LetLtxMacro{\LWR@origrmfamily}{\rmfamily}
791 \LetLtxMacro{\LWR@origsfamily}{\sffamily}
792 \LetLtxMacro{\LWR@origttfamily}{\ttfamily}
793 \LetLtxMacro{\LWR@origbfseries}{\bfseries}
794 \LetLtxMacro{\LWR@origmdseries}{\mdseries}
795 \LetLtxMacro{\LWR@origupshape}{\upshape}
796 \LetLtxMacro{\LWR@origslshape}{\slshape}
797 \LetLtxMacro{\LWR@origscshape}{\scshape}
798 \LetLtxMacro{\LWR@origitshape}{\itshape}
799 \LetLtxMacro{\LWR@origem}{\em}
800 \LetLtxMacro{\LWR@orignormalfont}{\normalfont}
801
802 \let\LWR@origraggedright\raggedright
803 \let\LWR@origonecolumn\onecolumn
804
805 \let\LWR@origsp\sp
806 \let\LWR@origsb\sb
807 \LetLtxMacro\LWR@origtextsuperscript\textsuperscript
808 \LetLtxMacro\LWR@orig@textsuperscript\@textsuperscript
809
810 \AtBeginDocument{
811 \LetLtxMacro\LWR@origtextsubscript\textsubscript
812 \LetLtxMacro\LWR@orig@textsubscript\@textsubscript
813 }
814
815 \LetLtxMacro\LWR@origunderline\underline
816
817 \let\LWR@orignewpage\newpage
818
819 \let\LWR@origpagestyle\pagestyle
820 \let\LWR@origthispagestyle\thispagestyle
821 \LetLtxMacro\LWR@origpagenumbering\pagenumbering
822
823 \LetLtxMacro{\LWR@origminipage}{\minipage}
824 \let\LWR@origendminipage\endminipage
825 \LetLtxMacro{\LWR@origparbox}{\parbox}
826
827 \let\LWR@orignewline\newline
828
829
830 \AtBeginDocument{% in case packages change definition
831 \let\LWR@orig@trivlist\@trivlist
832 \let\LWR@origtrivlist\trivlist
833 \let\LWR@origendtrivlist\endtrivlist
834 \LetLtxMacro\LWR@origitem\item
```

```

835 \LetLtxMacro\LWR@origitemize\itemize
836 \LetLtxMacro\LWR@endorigitemize\enditemize
837 \LetLtxMacro\LWR@origenumerate\enumerate
838 \LetLtxMacro\LWR@endorigenumerate\endenumerate
839 \LetLtxMacro\LWR@origdescription\description
840 \LetLtxMacro\LWR@endorigdescription\enddescription
841 \let\LWR@orig@mklab\@mklab
842 \let\LWR@origmakelabel\makelabel
843 \let\LWR@orig@donoparitem\@donoparitem
844 \LetLtxMacro\LWR@orig@item\@item
845 \let\LWR@orig@nbitem\@nbitem
846 }
847
848 \let\LWR@origpar\par
849
850 \LetLtxMacro{\LWR@origfootnote}{\footnote}
851 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
852
853 \let\LWR@origclearpage\clearpage
854
855
856 \AtBeginDocument{% in case packages change definition
857 \LetLtxMacro\LWR@orighline\hline%
858 \LetLtxMacro\LWR@origcline\cline%
859 }

860 \end{warpHTML}

```

33 Accents

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

for HTML output: 861 \begin{warpHTML}

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

```
862 \AtBeginDocument{
```

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

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

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

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

```

864 \ifPDFTeX
865 \LetLtxMacro{\LWR@origequalaccent}{\=}
866 \LetLtxMacro{\LWR@origdotaccent}{\cdot}
867 \LetLtxMacro{\LWR@origu}{\u}
868 \LetLtxMacro{\LWR@origv}{\v}
869 \LetLtxMacro{\LWR@origc}{\c}
870 \LetLtxMacro{\LWR@origd}{\d}
871 \LetLtxMacro{\LWR@origb}{\b}

```

The HTML redefinitions follow.

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

```

872 \renewcommand*{\=}[1]{\#1\HTMLUnicode{0305}}
873 \renewcommand*{\cdot}[1]{\#1\HTMLUnicode{0307}}
874 \renewcommand*{\u}[1]{\#1\HTMLUnicode{0306}}
875 \renewcommand*{\v}[1]{\#1\HTMLUnicode{030C}}
876 \renewcommand*{\c}[1]{\#1\HTMLUnicode{0327}}
877 \renewcommand*{\d}[1]{\#1\HTMLUnicode{0323}}
878 \renewcommand*{\b}[1]{\#1\HTMLUnicode{0331}}
879 \fi

```

For all engines, a Unicode diacritical tie is used:

```

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

```

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

```

882 \ifPDFTeX
883 \newcommand*{\LWR@restoreorigaccents}{%
884 \LetLtxMacro{\=}{\LWR@origequalaccent}%
885 \LetLtxMacro{\cdot}{\LWR@origdotaccent}%
886 \LetLtxMacro{\u}{\LWR@origu}%
887 \LetLtxMacro{\v}{\LWR@origv}%
888 \LetLtxMacro{\t}{\LWR@origt}%
889 \LetLtxMacro{\c}{\LWR@origc}%
890 \LetLtxMacro{\d}{\LWR@origd}%
891 \LetLtxMacro{\b}{\LWR@origb}%
892 }
893 \else% XeLaTeX, LuaLaTeX:
894 \newcommand*{\LWR@restoreorigaccents}{%
895 \LetLtxMacro{\t}{\LWR@origt}%
896 }
897 \fi
898 }% AtBeginDocument

```

```
899 \end{warpHTML}
```

34 Configuration Files

```
900 \begin{warpprint}
901 \typeout{lwarp: generating configuration files}
902 \end{warpprint}
```

34.1 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.

```
903 \begin{warpprint}
904 \immediate\openout\LWR@quickfile=\jobname_html.tex
905 \immediate\write\LWR@quickfile{%
906 \detokenize{\PassOptionsToPackage}%
907 {warpHTML,BaseJobname=\jobname}{lwarp}%
908 }
909 \immediate\write\LWR@quickfile{%
910 \detokenize{\input}\string{\jobname.tex}\string }%
911 }
912 \immediate\closeout\LWR@quickfile
913 \end{warpprint}
```

34.2 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:

```

914 \begin{warpprint}
915 \ifcsdef{LWR@quickfile}{\newwrite{LWR@quickfile}}
916 \immediate\openout{LWR@quickfile}=lwarpmk.conf
917 \ifbool{usingOSWindows}{
918 \immediate\write{LWR@quickfile}{opsystem = "Windows"}
919 }{
920 \immediate\write{LWR@quickfile}{opsystem = "Unix"}
921 }
922 \ifPDFTeX
923 \immediate\write{LWR@quickfile}{latexname = "pdflatex"}
924 \fi
925 \ifXeTeX
926 \immediate\write{LWR@quickfile}{latexname = "xelatex"}
927 \fi
928 \ifLuaTeX
929 \immediate\write{LWR@quickfile}{latexname = "lualatex"}
930 \fi
931 \immediate\write{LWR@quickfile}{sourcename = "\jobname"}
932 \immediate\write{LWR@quickfile}{%
933 homehtmlfilename = "\HomeHTMLFilename"%
934 }
935 \immediate\write{LWR@quickfile}{htmlfilename = "\HTMLFilename"}
936 \immediate\write{LWR@quickfile}{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
937 \immediate\write{LWR@quickfile}{language = "\LWR@IndexLanguage"}
938 \immediate\write{LWR@quickfile}{xdyfile = "\LWR@xdyFilename"}
939 \immediate\closeout{LWR@quickfile}
940 \end{warpprint}

```

34.3 project.lwarpmkconf

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

```

941 \begin{warpprint}
942 \ifcsdef{LWR@quickfile}{\newwrite{LWR@quickfile}}
943 \immediate\openout{LWR@quickfile}=\jobname.lwarpmkconf
944 \ifbool{usingOSWindows}{
945 \immediate\write{LWR@quickfile}{opsystem = "Windows"}
946 }{
947 \immediate\write{LWR@quickfile}{opsystem = "Unix"}
948 }
949 \ifPDFTeX
950 \immediate\write{LWR@quickfile}{latexname = "pdflatex"}
951 \fi
952 \ifXeTeX
953 \immediate\write{LWR@quickfile}{latexname = "xelatex"}
954 \fi
955 \ifLuaTeX

```

```

956 \immediate\write\LWR@quickfile{latexname = "lualatex"}
957 \fi
958 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
959 \immediate\write\LWR@quickfile{%
960 homehtmlfilename = "\HomeHTMLFilename"%
961 }
962 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
963 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}"}
964 \immediate\write\LWR@quickfile{language = "\LWR@IndexLanguage"}
965 \immediate\write\LWR@quickfile{xdyfile = "\LWR@xdyFilename"}
966 \immediate\closeout\LWR@quickfile
967 \end{warpprint}

```

34.4 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.

```

968 \begin{warpprint}
969 \begin{filecontents*}{lwarp.css}
970 /*
971  CSS stylesheet for the LaTeX lwarp package
972  Copyright 2016-2018 Brian Dunn -- BD Tech Concepts LLC
973 */
974
975
976 /* a fix for older browsers: */
977 header, section, footer, aside, nav, main,
978     article, figure { display: block; }
979
980
981 A:link {color:#000080 ; text-decoration: none ; }
982 A:visited {color:#800000 ; }
983 A:hover {color:#000080 ; text-decoration: underline ;}
984 A:active {color:#800000 ; }
985
986 a.tocpart {display: inline-block ; margin-left: 0em ;
987     font-weight: bold ;}
988 a.tocchapter {display: inline-block ; margin-left: 0em ;
989     font-weight: bold ;}
990 a.tocsection {display: inline-block ; margin-left: 1em ;
991     text-indent: -.5em ; font-weight: bold ;}
992 a.tocsubsection {display: inline-block ; margin-left: 2em ;
993     text-indent: -.5em ;}
994 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;

```

```
995     text-indent: -.5em ; }
996 a.tocparagraph {display: inline-block ; margin-left: 4em ;
997     text-indent: -.5em ; }
998 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
999     text-indent: -.5em ; }
1000 a.tocfigure {margin-left: 0em}
1001 a.tocsubfigure {margin-left: 2em}
1002 a.toctable {margin-left: 0em}
1003 a.tocsubtable {margin-left: 2em}
1004 a.toctheorem {margin-left: 0em}
1005 a.toclstlisting {margin-left: 0em}
1006
1007 body {
1008     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1009         "Lucida Bright", Georgia, serif;
1010     background: #FAF7F4 ;
1011     color: black ;
1012     margin: 0em ;
1013     padding: 0em ;
1014     font-size: 100% ;
1015     line-height: 1.2 ;
1016 }
1017
1018 p {margin: 1.5ex 0em 1.5ex 0em ;}
1019 table p {margin: .5ex 0em .5ex 0em ;}
1020
1021 /* Holds a section number to add space between it and the name */
1022 span.sectionnumber { margin-right: 0em }
1023
1024 /* Inserted in front of index lines */
1025 span.indexitem {margin-left: 0em}
1026 span.indexsubitem {margin-left: 2em}
1027 span.indexsubsubitem {margin-left: 4em}
1028
1029 div.hidden, span.hidden { display: none ; }
1030
1031 kbd {
1032     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1033         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1034         "Courier New", monospace;
1035     font-size: 100% ;
1036 }
1037
1038 pre { padding: 3pt ; }
1039
1040 span.strong { font-weight: bold; }
1041
1042 span.textmd { font-weight: normal; }
1043
1044 span.textsc { font-variant: small-caps; }
```

```
1045
1046 span.textsl { font-style: oblique; }
1047
1048 span.textup { font-variant: normal; }
1049
1050 span.textrm {
1051     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1052     "Lucida Bright", Georgia, serif;
1053 }
1054
1055 span.textsf {
1056     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1057     Geneva, Verdana, sans-serif ;
1058 }
1059
1060 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
1061
1062 span.underline {
1063     text-decoration: underline ;
1064     text-decoration-skip ;
1065 }
1066
1067
1068 /* For realscripts */
1069 .supsubscript {
1070     display: inline-block;
1071     text-align: left ;
1072 }
1073
1074 .supsubscript sup,
1075 .supsubscript sub {
1076     position: relative;
1077     display: block;
1078     font-size: .5em;
1079     line-height: 1;
1080 }
1081
1082 .supsubscript sup {
1083     top: .5em;
1084 }
1085
1086 .supsubscript sub {
1087     top: .5em;
1088 }
1089
1090 span.attribution {
1091     margin-left: 1em ; font-size: 80% ; font-variant: small-caps;
1092 }
1093
1094 span.citetitle {
```

```
1095 margin-left: 1em ; font-size: 80% ; font-style: oblique;
1096 }
1097
1098 span.poemtitle {
1099   font-size: 120% ; font-weight: bold;
1100 }
1101
1102 pre.tabbing {
1103   font-family: "Linux Libertine Mono O", "Lucida Console",
1104               "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1105               "Liberation Mono", "FreeMono", "Andale Mono",
1106               "Nimbus Mono L", "Courier New", monospace;
1107 }
1108
1109 blockquote {
1110   margin-left: 0px ;
1111   margin-right: 0px ;
1112 }
1113
1114 /* quotchap is for the quotchap package */
1115 div.quotchap {
1116   font-style: oblique ;
1117   overflow-x: auto ;
1118   margin-left: 2em ;
1119   margin-right: 2em ;
1120 }
1121
1122 blockquote p, div.quotchap p {
1123   line-height: 1.5;
1124   text-align: left ;
1125   font-size: .85em ;
1126   margin-left: 3em ;
1127   margin-right: 3em ;
1128 }
1129
1130 /* qauthor is for the quotchap package */
1131 div.qauthor {
1132   display: block ;
1133   text-align: right ;
1134   margin-left: auto ;
1135   margin-right: 2em ;
1136   font-size: 80% ;
1137   font-variant: small-caps;
1138 }
1139
1140 div.qauthor p {
1141   text-align: right ;
1142 }
1143
1144 blockquotation {
```

```
1145     margin-left: 0px ;
1146     margin-right: 0px ;
1147 }
1148
1149 blockquotation p {
1150     line-height: 1.5;
1151     text-align: left ;
1152     font-size: .85em ;
1153     margin-left: 3em ;
1154     margin-right: 3em ;
1155 }
1156
1157 div.epigraph, div.dictum {
1158     line-height: 1.2;
1159     text-align: left ;
1160     padding: 3ex 1em 0ex 1em ;
1161     /*     margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
1162     margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
1163     /*     margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
1164     font-size: .85em ;
1165     max-width: 27em ;
1166 }
1167
1168
1169
1170 div.epigraphsource, div.dictumauthor {
1171     text-align:right ;
1172     margin-left:auto ;
1173     /*     max-width: 50% ; */
1174     border-top: 1px solid #A0A0A0 ;
1175     padding-bottom: 3ex ;
1176     line-height: 1.2;
1177 }
1178
1179 div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
1180 div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1181 div.dictumauthor { font-style:italic }
1182
1183
1184 /* lettrine package: */
1185 span.lettrine { font-size: 3ex ; float: left ; }
1186 span.lettrinetext { font-variant: small-caps ; }
1187
1188 /* ulem and soul packages: */
1189 span.uline {
1190     text-decoration: underline ;
1191     text-decoration-skip ;
1192 }
1193
1194 span.uline {
```

```
1195     text-decoration: underline ;
1196     text-decoration-skip ;
1197     text-decoration-style: double ;
1198 }
1199
1200 span.uwave {
1201     text-decoration: underline ;
1202     text-decoration-skip ;
1203     text-decoration-style: wavy ;
1204 }
1205
1206 span.sout {
1207     text-decoration: line-through ;
1208 }
1209
1210 span.xout {
1211     text-decoration: line-through ;
1212 }
1213
1214 span.dashuline {
1215     text-decoration: underline ;
1216     text-decoration-skip ;
1217     text-decoration-style: dashed ;
1218 }
1219
1220 span.dotuline {
1221     text-decoration: underline ;
1222     text-decoration-skip ;
1223     text-decoration-style: dotted ;
1224 }
1225
1226 span.letterspacing { letter-spacing: .2ex ; }
1227
1228 span.capsspacing {
1229     font-variant: small-caps ;
1230     letter-spacing: .1ex ;
1231 }
1232
1233 span.highlight { background: #F8E800 ; }
1234
1235
1236
1237
1238 html body {
1239     margin: 0 ;
1240     line-height: 1.2;
1241 }
1242
1243
1244 body div {
```

```
1245 margin: 0ex;
1246 }
1247
1248
1249 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1250 {
1251     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1252         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1253         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1254         "Times New Roman", serif;
1255     font-style: normal ;
1256     font-weight: bold ;
1257     text-align: left ;
1258 }
1259
1260 h1 { /* title of the entire website, used on each page */
1261     text-align: center ;
1262     font-size: 2.5em ;
1263     padding: .4ex 0em 0ex 0em ;
1264 }
1265 h2 { font-size: 2.25em }
1266 h3 { font-size: 2em }
1267 h4 { font-size: 1.75em }
1268 h5 { font-size: 1.5em }
1269 h6 { font-size: 1.25em }
1270 span.paragraph {font-size: 1em ; font-variant: normal ;
1271     margin-right: 1em ; }
1272 span.subparagraph {font-size: 1em ; font-variant: normal ;
1273     margin-right: 1em ; }
1274
1275 div.minisec {
1276     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1277         Geneva, Verdana, sans-serif ;
1278     font-style: normal ;
1279     font-weight: bold ;
1280     text-align: left ;
1281 }
1282
1283 /* Title of the file */
1284 h1 {
1285     margin: 0ex 0em 0ex 0em ;
1286     line-height: 1.3;
1287     text-align: center ;
1288 }
1289
1290 /* Part */
1291 h2 {
1292     margin: 1ex 0em 1ex 0em ;
1293     line-height: 1.3;
1294     text-align: center ;
```

```
1295 }
1296
1297 /* Chapter */
1298 h3 {
1299     margin: 3ex 0em 1ex 0em ;
1300     line-height: 1.3;
1301 }
1302
1303 /* Section */
1304 h4 {
1305     margin: 3ex 0em 1ex 0em ;
1306     line-height: 1.3;
1307 }
1308
1309 /* Sub-Section */
1310 h5 {
1311     margin: 3ex 0em 1ex 0em ;
1312     line-height: 1.3;
1313 }
1314
1315 /* Sub-Sub-Section */
1316 h6 {
1317     margin: 3ex 0em 1ex 0em ;
1318     line-height: 1.3;
1319 }
1320
1321
1322 div.titlepage {
1323     text-align: center ;
1324 }
1325
1326 .footnotes {
1327     font-size: .85em ;
1328     margin: 3ex 1em 0ex 1em ;
1329     padding-bottom: 1ex ;
1330     border-top: 1px solid silver ;
1331 }
1332
1333 .marginpar, .marginparblock {
1334     max-width: 50%;
1335     float: right;
1336     text-align: left;
1337     margin: 1ex 0.5em 1ex 1em ;
1338     padding: 1ex 0.5em 1ex 0.5em ;
1339     font-size: 85% ;
1340     border-top: 1px solid silver ;
1341     border-bottom: 1px solid silver ;
1342     overflow-x: auto;
1343 }
1344
```

```
1345 .marginpar br { margin-bottom: 2ex ; }
1346
1347 div.marginblock, div.marginparblock {
1348     max-width:50%;
1349     float:right;
1350     text-align:left;
1351     margin: 1ex 0.5em 1ex 1em ;
1352     padding: 1ex 0.5em 1ex 0.5em ;
1353     overflow-x: auto;
1354 }
1355
1356 div.marginblock div.minipage,
1357 div.marginparblock div.minipage {
1358     display: block ;
1359     margin: 0pt auto 0pt auto ;
1360 }
1361
1362 div.marginblock div.minipage p ,
1363 div.marginparblock div.minipage p
1364     { font-size: 85%}
1365
1366 div.marginblock br ,
1367 div.marginparblock br
1368     { margin-bottom: 2ex ; }
1369
1370
1371 section.textbody div.footnotes{
1372     margin: 3ex 0em 0ex 0em ;
1373     border-bottom: 2px solid silver ;
1374 }
1375
1376 .footnoteheader {
1377     border-top: 2px solid silver ;
1378     margin-top: 3ex ;
1379     padding-top: 1ex ;
1380     font-weight: bold ;
1381 }
1382
1383 .mpfootnotes {
1384     text-align: left ;
1385     font-size: .85em ;
1386     margin-left: 1em ;
1387     border-top: 1px solid silver ;
1388 }
1389
1390 /* Remove footnote top border in the title page. */
1391 div.titlepage div.mpfootnotes {
1392     border-top: none ;
1393 }
1394
```

```
1395
1396
1397 ol {
1398     margin: 1ex 1em 1ex 0em;
1399     line-height: 1.2;
1400 }
1401
1402 ul, body dir, body menu {
1403     margin: 3ex 1em 3ex 0em;
1404     line-height: 1.2;
1405 }
1406
1407 li { margin: 0ex 0em 1ex 0em; }
1408
1409 html {
1410     margin: 0;
1411     padding: 0;
1412 }
1413
1414 .programlisting {
1415     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1416                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1417                 "Courier New", monospace;
1418     margin: 1ex 0ex 1ex 0ex ;
1419     padding: .5ex 0pt .5ex 0pt ;
1420     overflow-x: auto;
1421 }
1422
1423 section.textbody>pre.programlisting {
1424 border-top: 1px solid silver ;
1425 border-bottom: 1px solid silver ;
1426 }
1427
1428
1429 .inlineprogramlisting {
1430     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1431                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1432                 "Courier New", monospace;
1433     overflow-x: auto;
1434 }
1435
1436 span.listinglabel {
1437     display: inline-block ;
1438     font-size: 70% ;
1439     width: 4em ;
1440     text-align: right ;
1441     margin-right: 2em ;
1442 }
1443
1444 div.abstract {
```

```
1445 margin: 2em 5% 2em 5% ;
1446 padding: 1ex 1em 1ex 1em ;
1447 /* font-weight: bold ; */
1448 font-size: 90% ;
1449 text-align: left ;
1450 }
1451
1452 div.abstract dl {line-height:1.5;}
1453 div.abstract dt {color:#304070;}
1454
1455 div.abstracttitle{
1456     font-family: "URW Classico", Optima, "Linux Biolinum O",
1457         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1458         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1459     font-weight:bold;
1460     font-size:1.25em;
1461     text-align: center ;
1462 }
1463
1464 span.abstracrunintitle{
1465     font-family: "URW Classico", Optima, "Linux Biolinum O",
1466         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1467         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1468     font-weight:bold;
1469 }
1470
1471
1472 .verbatim {
1473     overflow-x: auto ;
1474 }
1475
1476 .alltt {
1477     overflow-x: auto ;
1478 }
1479
1480
1481 .bverbatim {
1482     margin: 1ex 0pt 1ex 0pt ;
1483     padding: .5ex 0pt .5ex 0pt ;
1484     overflow-x: auto ;
1485 }
1486
1487 .lverbatim {
1488     margin: 1ex 0pt 1ex 0pt ;
1489     padding: .5ex 0pt .5ex 0pt ;
1490     overflow-x: auto ;
1491 }
1492
1493 .fancyvrb {
1494     font-size:.85em ;
```

```
1495     margin: 3ex 0pt 3ex 0pt
1496 }
1497
1498 .fancyvrblabel {
1499     font-weight:bold;
1500     text-align: center ;
1501 }
1502
1503
1504 .verse {
1505     font-family: "Linux Libertine Mono O", "Lucida Console",
1506                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1507                 "Liberation Mono", "FreeMono", "Andale Mono",
1508                 "Nimbus Mono L", "Courier New", monospace;
1509     margin-left: 1em ;
1510 }
1511
1512
1513 div.singlespace { line-height: 1.2 ; }
1514 div.onehalfspace { line-height: 1.5 ; }
1515 div.doublespace { line-height: 2 ; }
1516
1517
1518 /* Word processor format output: */
1519 div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
1520 div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
1521 div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ;}
1522
1523
1524
1525
1526 /* Minipage environments, vertically aligned to top, center, bottom: */
1527 .minipage, .fminipage, .fcolorminipage {
1528     /* display: inline-block ; */
1529     /* Mini pages which follow each other will be tiled. */
1530     margin: .25em .25em .25em .25em;
1531     padding: .25em .25em .25em .25em;
1532     display: inline-flex;
1533     flex-direction: column ;
1534     overflow: auto;
1535 }
1536
1537 /* Paragraphs in the flexbox did not collapse their margins. */
1538 /* Have not yet researched this. */
1539 .minipage p {margin: .75ex 0em .75ex 0em ;}
1540
1541 .fboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1542 .fcolorbox .minipage, .fcolorboxBlock .minipage
1543     {border: none ; background: none;}
1544
```

```
1545 .fbox, .fboxBlock { border: 1px solid black ; }
1546
1547 .fbox, .fboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1548 .fminipage, .fcolorminipage
1549     {display: inline-block}
1550
1551 .shadowbox, .shabox {
1552     border: 1px solid black;
1553     box-shadow: 3px 3px 3px #808080 ;
1554     border-radius: 0px ;
1555     padding: .4ex .3em .4ex .3em ;
1556     margin: 0pt .3ex 0pt .3ex ;
1557     display: inline-block ;
1558 }
1559
1560 .doublebox {
1561     border: 3px double black;
1562     border-radius: 0px ;
1563     padding: .4ex .3em .4ex .3em ;
1564     margin: 0pt .3ex 0pt .3ex ;
1565     display: inline-block ;
1566 }
1567
1568 .ovalbox, .Ovalbox {
1569     border: 1px solid black;
1570     border-radius: 1ex ;
1571     padding: .4ex .3em .4ex .3em ;
1572     margin: 0pt .3ex 0pt .3ex ;
1573     display: inline-block ;
1574 }
1575
1576 .Ovalbox { border-width: 2px ; }
1577
1578 .framebox {
1579     border: 1px solid black;
1580     border-radius: 0px ;
1581     padding: .3ex .2em 0ex .2em ;
1582     margin: 0pt .1ex 0pt .1ex ;
1583     display: inline-block ;
1584 }
1585
1586
1587 .mdframed {
1588 /*     padding: 0ex ; */
1589 /*     border: 1px solid black; */
1590 /*     border-radius: 0px ; */
1591     padding: 0ex ;
1592     margin: 3ex 5% 3ex 5% ;
1593 /*     display: inline-block ; */
1594 }
```

```
1595
1596 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1597
1598 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1599
1600 .mdframedtitle {
1601     padding: .5em ;
1602     display: block ;
1603     font-size: 130% ;
1604     margin-bottom: 1ex ;
1605 }
1606
1607 .mdframedsubtitle {
1608     padding: 0ex .5em 0ex .5em ;
1609     display: block ;
1610     font-size: 115% ;
1611 }
1612
1613 .mdframedsubsubtitle {
1614     padding: 0ex .5em 0ex .5em ;
1615     display: block ;
1616 }
1617
1618 .mdtheorem {
1619     padding: 0ex .5em 0ex .5em ;
1620     margin: 3ex 5% 3ex 5% ;
1621 /*    display: inline-block ; */
1622 }
1623
1624
1625 /* framed package */
1626 .framed, pre.boxedverbatim, fcolorbox {
1627     margin: 3ex 0em 3ex 0em ;
1628     border: 1px solid black;
1629     border-radius: 0px ;
1630     padding: .3ex 1em 0ex 1em ;
1631     display: block ;
1632 }
1633
1634 .shaded {
1635     margin: 3ex 0em 3ex 0em ;
1636     padding: .3ex 1em .3ex 1em ;
1637     display: block ;
1638 }
1639
1640 .snugframed {
1641     margin: 3ex 0em 3ex 0em ;
1642     border: 1px solid black;
1643     border-radius: 0px ;
1644     display: block ;
```

```
1645 }
1646
1647 .framedleftbar {
1648     margin: 3ex 0em 3ex 0em ;
1649     border-left: 3pt solid black;
1650     border-radius: 0px ;
1651     padding: .3ex .2em .3ex 1em ;
1652     display: block ;
1653 }
1654
1655 .framedtitle {
1656     margin: 0em ;
1657     padding: 0em ;
1658     font-size: 130%
1659 }
1660
1661 .framedtitle p { padding: .3em }
1662
1663
1664
1665 dl {
1666     margin: 1ex 2em 1ex 0em;
1667     line-height: 1.3;
1668 }
1669
1670 dl dt {
1671     margin-top: 1ex;
1672     margin-left: 1em ;
1673     font-weight: bold;
1674 }
1675
1676 dl dd p { margin-top: 0em; }
1677
1678
1679 nav {
1680     font-family: "URW Classico", Optima, "Linux Biolinum O",
1681         "DejaVu Sans", "Bitstream Vera Sans",
1682         Geneva, Verdana, sans-serif ;
1683     margin-bottom: 4ex ;
1684 }
1685
1686 nav p {
1687     line-height: 1.2 ;
1688     margin-top:.5ex ;
1689     margin-bottom:.5ex;
1690     font-size: .9em ;
1691 }
1692
1693
1694
```

```
1695 img, img.hyperimage, img.borderimage {
1696     max-width: 600px;
1697     border: 1px solid silver;
1698     box-shadow: 3px 3px 3px #808080 ;
1699     padding: .5% ;
1700     margin: .5% ;
1701     background: none ;
1702 }
1703
1704 img.inlineimage{
1705     padding: 0px ;
1706     box-shadow: none ;
1707     border: none ;
1708     background: none ;
1709     margin: 0px ;
1710     display: inline-block ;
1711     border-radius: 0px ;
1712 }
1713
1714 img.logoimage{
1715     max-width: 300px ;
1716     box-shadow: 3px 3px 3px #808080 ;
1717     border: 1px solid black ;
1718     background:none ;
1719     padding:0 ;
1720     margin:.5ex ;
1721     border-radius: 10px ;
1722 }
1723
1724
1725 .section {
1726 /*
1727     To have each section float relative to each other:
1728 */
1729 /*
1730     display: block ;
1731     float: left ;
1732     position: relative ;
1733     background: white ;
1734     border: 1px solid silver ;
1735     padding: .5em ;
1736 */
1737     margin: 0ex .5em 0ex .5em ;
1738     padding: 0 ;
1739 }
1740
1741
1742 figure {
1743     margin: 3ex auto 3ex auto ;
1744     padding: 1ex 1em 1ex 1em ;
```

```
1745     overflow-x: auto ;
1746 }
1747
1748
1749 /* To automatically center images in figures: */
1750 /*
1751 figure img.inlineimage {
1752     margin: 0ex auto 0ex auto ;
1753     display: block ;
1754 }
1755 */
1756
1757 /* To automatically center minipages in figures: */
1758 /*
1759 figure div.minipage, figure div.minipage div.minipage {
1760     margin: 1ex auto 1ex auto ;
1761     display: block ;
1762 }
1763 */
1764
1765 figure div.minipage p { font-size: 85% ; }
1766
1767 figure.subfigure, figure.subtable {
1768     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1769 }
1770
1771 figcaption .minipage { margin:0 ; padding: 0 }
1772
1773 div.minipage figure { border: none ; box-shadow: none ; }
1774
1775 div.floatrow { text-align: center; }
1776
1777 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1778
1779 div.floatfoot { font-size: .85em ;
1780     border-top: 1px solid silver ; line-height: 1.2 ; }
1781
1782 figcaption , .lstlistingtitle {
1783     font-size: .85em ;
1784     text-align: center ;
1785     font-weight: bold ;
1786     margin-top: 1ex ;
1787     margin-bottom: 1ex ;
1788 }
1789
1790 figure.subfigure figcaption, figure.subtable figcaption {
1791     border-bottom: none ; background: none ;
1792 }
1793
1794 div.nonfloatcaption {
```

```

1795     margin: 1ex auto 1ex auto ;
1796     font-size: .85em ;
1797     text-align: center ;
1798     font-weight: bold ;
1799 }
1800
1801 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1802 figure div.floatrow div.minipage figcaption {
1803     border: none ;
1804     background: none ;
1805 }
1806
1807
1808 table {
1809     margin: 1ex auto 1ex auto ;
1810     border-collapse: separate ;
1811     border-spacing: 0px ;
1812     line-height: 1.3 ;
1813 }
1814
1815 tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
1816     margin-bottom: 0ex ; } /* for \hline */
1817
1818 tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
1819     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
1820
1821 td {padding: .5ex .5em .5ex .5em ;}
1822
1823 table td.tdl { text-align: left ; vertical-align: middle ; }
1824 table td.tdc { text-align: center ; vertical-align: middle ; }
1825 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
1826 table td.tdbang { text-align: center ; vertical-align: middle ; }
1827 table td.tdr { text-align: right ; vertical-align: middle ; }
1828 table td.tdp { text-align: left ; vertical-align: bottom ; }
1829 table td.tdm { text-align: left ; vertical-align: middle ; }
1830 table td.tdb { text-align: left ; vertical-align: top ; }
1831 table td.tdP { text-align: center ; vertical-align: bottom ; }
1832 table td.tdM { text-align: center ; vertical-align: middle ; }
1833 table td.tdB { text-align: center ; vertical-align: top ; }
1834
1835 table td.tvertbarl { border-left: 1px solid black }
1836 table td.tvertbarr { border-right: 1px solid black }
1837
1838
1839 /* for cmidrules: */
1840 table td.tdrule {
1841     border-top: 1px solid #A0A0A0 ;
1842 }
1843
1844 table td.tdrulel {

```

```
1845     border-top-left-radius:.5em ;
1846     border-top: 1px solid #A0A0A0 ;
1847 }
1848
1849 table td.tdruler {
1850     border-top-right-radius:.5em ;
1851     border-top: 1px solid #A0A0A0 ;
1852 }
1853
1854 table td.tdrulelr {
1855     border-top-left-radius:.5em ;
1856     border-top-right-radius:.5em ;
1857     border-top: 1px solid #A0A0A0 ;
1858 }
1859
1860
1861 /* Margins of paragraphs inside table cells: */
1862 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
1863     padding-bottom: 1ex ; margin: 0ex ; }
1864 td.tdm p , td.tmbrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
1865     padding-bottom: 1ex ; margin: 0ex ; }
1866 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
1867     padding-bottom: 1ex ; margin: 0ex ; }
1868
1869 td.tdp , td.tdprule , td.tdP , td.tdPrule
1870     { padding: 0ex .5em 0ex .5em ; }
1871 td.tdm , td.tdmrule , td.tdM , td.tdMrule
1872     { padding: 0ex .5em 0ex .5em ; }
1873 td.tdb , td.tdbrule , td.tdB , td.tdBrule
1874     { padding: 0ex .5em 0ex .5em ; }
1875
1876
1877 /* table notes: */
1878 .tnotes {
1879     margin: 0ex 5% 1ex 5% ;
1880     padding: 0.5ex 1em 0.5ex 1em;
1881     font-size:.85em;
1882     text-align: left ;
1883 }
1884
1885 .tnotes dl dt p {margin-bottom:0px;}
1886
1887 .tnoteitemheader {margin-right: 1em;}
1888
1889
1890 /* for colortbl and cell color */
1891 div.cellcolor {
1892     width: 100% ;
1893     padding: .5ex .5em .5ex .5em ;
1894     margin: -.5ex -.5em -.5ex -.5em ;
```

```
1895 }
1896
1897
1898 /* for bigdelim */
1899 .ldelim, .rdelim { font-size: 200% }
1900
1901
1902 /* center, flushleft, flushright environments */
1903 div.center{text-align:center;}
1904 div.center table {margin-left:auto;margin-right:auto;}
1905 div.flushleft{text-align:left;}
1906 div.flushleft table {margin-left:0em ; margin-right:auto;}
1907 div.flushright{text-align:right;}
1908 div.flushright table {margin-left:auto ; margin-right: 0em ;}
1909
1910
1911 /* Fancybox */
1912 div.Btrivlist table tr td { padding: .2ex 0em ; }
1913
1914
1915 /* program listing callouts: */
1916 span.callout {
1917     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1918     Geneva, Verdana, sans-serif ;
1919     border-radius: .5em;
1920     background-color:black;
1921     color:white;
1922     padding:0px .25em 0px .25em;
1923     margin: 0 ;
1924     font-weight: bold;
1925     font-size:.72em ;
1926 }
1927
1928 div.programlisting pre.verbatim span.callout{
1929     font-size: .85em ;
1930 }
1931
1932
1933
1934
1935
1936 div.published
1937 {
1938     text-align: center ;
1939     font-variant: normal ;
1940     font-style: italic ;
1941     font-size: 1em ;
1942     margin: 3ex 0em 3ex 0em ;
1943 }
1944
```

```
1945 div.subtitle
1946 {
1947     text-align: center ;
1948     font-variant: normal ;
1949     font-style: italic ;
1950     font-size: 1.25em ;
1951     margin: 3ex 0em 3ex 0em ;
1952 }
1953
1954 div.subtitle p { margin: 1ex ; }
1955
1956 div.author
1957 {
1958     font-variant: normal ;
1959     font-style: normal ;
1960     font-size: 1em ;
1961     margin: 3ex 0em 3ex 0em ;
1962 }
1963
1964 div.oneauthor {
1965     display: inline-block ;
1966     margin: 3ex 1em 0ex 1em ;
1967 }
1968
1969 /*
1970 div.author table {
1971     margin: 3ex auto 0ex auto ;
1972     background: none ;
1973 }
1974
1975 div.author table tbody tr td { padding: .25ex ; }
1976 */
1977
1978 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1979
1980 div.titledate {
1981     text-align: center ;
1982     font-size: .85em ;
1983     font-style: italic;
1984     margin: 6ex 0em 6ex 0em ;
1985 }
1986
1987
1988 nav.topnavigation{
1989     text-align: left ;
1990     padding: 0.5ex 1em 0.5ex 1em ;
1991 /*     margin: 2ex 0em 3ex 0em ; */
1992     margin: 0 ;
1993     border-bottom: 1px solid silver ;
1994     border-top: 1px solid silver ;
```

```
1995     clear:right ;
1996 }
1997
1998 nav.botnavigation{
1999     text-align: left ;
2000     padding: 0.5ex 1em 0.5ex 1em ;
2001 /*     margin: 3ex 0em 2ex 0em ; */
2002     margin: 0 ;
2003     border-top: 1px solid silver ;
2004     border-bottom: 1px solid silver ;
2005     clear:right ;
2006 }
2007
2008
2009 header{
2010     line-height: 1.2 ;
2011     font-size: 1em ;
2012 /*     border-bottom: 2px solid silver ; */
2013     margin: 0px ;
2014     padding: 0ex 1em 0ex 1em ;
2015     text-align:center ;
2016 }
2017
2018 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
2019
2020
2021 footer{
2022     font-size: .85em ;
2023     line-height: 1.2 ;
2024     margin-top: 1ex ;
2025     border-top: 2px solid silver ;
2026     padding: 2ex 1em 2ex 1em ;
2027     clear:right ;
2028     text-align:left ;
2029 }
2030
2031
2032 a.linkhome { font-weight:bold ; font-size: 1em ;}
2033
2034
2035 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
2036
2037 img.lateximage{
2038     padding: 0px 0px 0px 0px ;
2039     box-shadow: none ;
2040     border: none ;
2041     background: none ;
2042     margin: 0px 0px -.15ex 0px ;
2043     /* pdfcrop leaves a slight margin, adjust to baseline */
2044     max-width: 100% ;
```

```
2045     border-radius: 0ex ;
2046     border: none ;
2047 }
2048
2049
2050
2051 nav.sidetoc {
2052     font-family: "DejaVu Serif", "Bitstream Vera Serif",
2053         "Lucida Bright", Georgia, serif;
2054     float:right ;
2055     width: 20%;
2056     border-left: 1px solid silver;
2057     border-top: 1px solid silver;
2058     border-bottom: 1px solid silver;
2059 /*     border-top: 2px solid #808080 ; */
2060     background: #FAF7F4 ;
2061     padding: 2ex 0em 2ex 1em ;
2062     margin: 0ex 0em 2ex 1em ;
2063     font-size:.9em ;
2064     border-radius: 20px 0px 0px 20px ;
2065 }
2066
2067 div.sidetoccontents {
2068 /*     border-top: 1px solid silver ; */
2069     overflow-y: auto ;
2070     width: 100% ;
2071     text-align: left ;
2072 }
2073
2074
2075 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
2076     text-indent: 0 ; }
2077
2078 nav.sidetoc p a {color:black ; font-size: .7em ;}
2079
2080 div.sidetoctitle {font-size: 1.2em; font-weight:bold; text-align:center;
2081     border-bottom: 1px solid silver ;    }
2082
2083 nav.sidetoc a:hover {text-decoration: underline ; }
2084
2085
2086
2087 section.textbody { margin: 0ex 1em 0ex 1em ;}
2088
2089
2090 div.multicolsheading { -webkit-column-span: all;
2091     -moz-column-span: all; column-span: all; }
2092 div.multicols { -webkit-columns: 3 380px ;
2093     -moz-columns: 3 380px ; columns: 3 380px ; }
2094 div.multicols p {margin-top: 0ex}
```

```
2095
2096
2097
2098 /* Used to support algorithmicx: */
2099 span.floatright { float: right ; }
2100
2101
2102
2103
2104 /* Native LaTeX theorems: */
2105
2106 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
2107 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
2108
2109
2110 /* theorem, amsthm, and ntheorem packages */
2111
2112 span.theoremheader,
2113 span.theoremheaderplain,
2114 span.theoremheaderdefinition,
2115 span.theoremheaderbreak,
2116 span.theoremheadermarginbreak,
2117 span.theoremheaderchangebreak,
2118 span.theoremheaderchange,
2119 span.theoremheadermargin
2120 {
2121     font-style:normal ; font-weight: bold ; margin-right: 1em ;
2122 }
2123
2124 span.amsthmnameplain,
2125 span.amsthmnamedefinition,
2126 span.amsthmnumberplain,
2127 span.amsthmnumberdefinition
2128 {
2129     font-style:normal ; font-weight: bold ;
2130 }
2131
2132
2133 span.amsthmnameremark,
2134 span.amsthmnumberremark
2135 {font-style:italic ; font-weight: normal ; }
2136
2137
2138 span.amsthmnoteplain,
2139 span.amsthmnotedefinition
2140 {font-style:normal ;}
2141
2142
2143 span.theoremheaderremark,
2144 span.theoremheaderproof,
```

```
2145 span.amsthmproofname
2146 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
2147
2148 span.theoremheadersc
2149 {
2150     font-style:normal ;
2151     font-variant: small-caps ;
2152     font-weight: normal ;
2153     margin-right: 1em ;
2154 }
2155
2156 .theoremendmark {float:right}
2157
2158 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
2159 div.theorembodybreak, div.theorembodynonumberbreak,
2160 div.theorembodymarginbreak,
2161 div.theorembodychangebreak,
2162 div.theorembodychange,
2163 div.theorembodymargin
2164 {
2165     font-style:italic;
2166     margin-top: 3ex ; margin-bottom: 3ex ;
2167 }
2168
2169 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
2170 div.theorembodyplainupright, nonumberplainuprightsc,
2171 div.amsthmbodydefinition, div.amsthmbodyremark,
2172 div.amsthmproof
2173 {
2174     font-style: normal ;
2175     margin-top: 3ex ; margin-bottom: 3ex ;
2176 }
2177
2178 span.amsthmnoteremark {}
2179
2180
2181
2182 /*
2183 For CSS LaTeX and related logos:
2184 Based on:
2185 http://edward.oconnor.cx/2007/08/tex-poshlet
2186 http://nitens.org/taraborelli/texlogo
2187 */
2188
2189 .latexlogofont {
2190     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2191         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2192     font-variant: normal ;
2193 }
2194
```

```
2195 .latexlogo {
2196     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2197         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2198     letter-spacing: .03em ;
2199     font-size: 1.1em;
2200 }
2201
2202 .latexlogo sup {
2203     text-transform: uppercase;
2204     letter-spacing: .03em ;
2205     font-size: 0.85em;
2206     vertical-align: 0.15em;
2207     margin-left: -0.36em;
2208     margin-right: -0.15em;
2209 }
2210
2211 .latexlogo sub {
2212     text-transform: uppercase;
2213     vertical-align: -0.5ex;
2214     margin-left: -0.1667em;
2215     margin-right: -0.125em;
2216     font-size: 1em;
2217 }
2218
2219 .xetexlogo {
2220     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2221         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2222     letter-spacing: .03em ;
2223     font-size: 1.1em;
2224 }
2225
2226 /* A smaller gap between Xe and Tex v.s. LaTeX: */
2227 .xetexlogo sub {
2228     text-transform: uppercase;
2229     vertical-align: -0.5ex;
2230     margin-left: -0.0667em;
2231     margin-right: -0.2em;
2232     font-size: 1em;
2233     letter-spacing: .03em ;
2234 }
2235
2236 /* A large gap between Xe and LaTeX v.s. TeX: */
2237 .xelatexlogo sub {
2238     text-transform: uppercase;
2239     vertical-align: -0.5ex;
2240     margin-left: -0.0667em;
2241     margin-right: -.05em;
2242     font-size: 1em;
2243     letter-spacing: .03em ;
2244 }
```

```
2245
2246 .amslogo {
2247     font-family: "TeXGyreChorus","URW Chancery L",
2248         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
2249         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
2250         "Hoefler Text", Times, "Times New Roman", serif;
2251     font-style: italic;
2252 }
2253
2254 .lyxlogo {
2255     font-family: "URW Classico", Optima, "Linux Biolinum O",
2256         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2257         Verdana, sans-serif ;
2258 }
2259
2260
2261
2262
2263 /* Only display top and bottom navigation if a small screen: */
2264 /* Hide the sidetoc if a small screen: */
2265 nav.topnavigation { display:none; }
2266 nav.botnavigation { display:none; }
2267
2268 @media screen and (max-width: 45em) {
2269 /*     nav.sidetoc {display:none;} */
2270     nav.sidetoc {
2271         float: none ;
2272         width: 100% ;
2273         margin: 5ex 0px 5ex 0px ;
2274         padding: 0 ;
2275         border-radius: 0 ;
2276         border-bottom: 1px solid black ;
2277         border-top: 1px solid black ;
2278         box-shadow: none ;
2279     }
2280 /*     nav.topnavigation { display:block } */
2281     nav.botnavigation { display:block }
2282     .marginpar {
2283         max-width: 100%;
2284         float: none;
2285         display:block ;
2286         margin: 1ex 1em 1ex 1em ;
2287     }
2288 }
2289
2290 @media print {
2291     body {
2292         font-family: "Linux Libertine O",
2293             "DejaVu Serif", "Bitstream Vera Serif",
2294             "Liberation Serif", "Nimbus Roman No 9 L",
```

```

2295         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2296     }
2297     nav.sidetoc { display:none; }
2298     nav.topnavigation { display: none; }
2299     nav.botnavigation { display: none; }
2300 }
2301
2302 @media handheld {
2303     nav.sidetoc { display:none; }
2304     nav.topnavigation { display:block }
2305     nav.botnavigation { display:block }
2306 }
2307
2308 @media projection {
2309     nav.sidetoc { display:none; }
2310     nav.topnavigation { display:block }
2311     nav.botnavigation { display:block }
2312 }
2313 \end{filecontents*}
2314 % \end{Verbatim}% for syntax highlighting
2315 \end{warpprint}

```

34.5 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.

```

2316 \begin{warpprint}
2317 \begin{filecontents*}{lwarp_sagebrush.css}
2318 @import url("lwarp.css") ;
2319
2320
2321 A:link {color:#105030 ; text-decoration: none ; }
2322 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2323 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2324 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2325
2326
2327
2328 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2329 {
2330     font-family: "URW Classico", Optima, "Linux Biolinum O",
2331         "Linux Libertine O", "Liberation Serif",
2332         "Nimbus Roman No 9 L", "FreeSerif",
2333         "Hoefler Text", Times, "Times New Roman", serif;

```

```
2334     font-variant: small-caps ;
2335     font-weight: normal ;
2336     color: #304070 ;
2337     text-shadow: 2px 2px 3px #808080;
2338 }
2339
2340 h1 {      /* title of the entire website, used on each page */
2341     font-variant: small-caps ;
2342     color: #304070 ;
2343     text-shadow: 2px 2px 3px #808080;
2344     background-color: #F7F7F0 ;
2345     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2346 }
2347
2348 h1 {
2349     border-bottom: 1px solid #304070;
2350     border-top: 2px solid #304070;
2351 }
2352
2353 h2 {
2354     border-bottom: 1px solid #304070;
2355     border-top: 2px solid #304070;
2356     background-color: #F7F7F0 ;
2357     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2358 }
2359
2360
2361
2362 div.abstract {
2363     background: #f5f5eb ;
2364     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2365
2366     border: 1px solid silver;
2367     border-radius: 1em ;
2368 }
2369
2370 div.abstract dl {line-height:1.5;}
2371 div.abstract dt {color:#304070;}
2372
2373 div.abstracttitle{
2374     font-family: "URW Classico", Optima, "Linux Biolinum O",
2375         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2376         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2377     font-weight:bold;
2378     font-variant: small-caps ;
2379     font-size:1.5em;
2380     border-bottom: 1px solid silver ;
2381     color: #304070 ;
2382     text-align: center ;
2383     text-shadow: 1px 1px 2px #808080;
```

```
2384 }
2385
2386 span.abstractrunintitle{
2387     font-family: "URW Classico", Optima, "Linux Biolinum O",
2388         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2389         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2390     font-weight:bold;
2391 }
2392
2393
2394 div.epigraph, div.dictum {
2395     background: #f5f5eb ;
2396     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2397
2398     border: 1px solid silver ;
2399     border-radius: 1ex ;
2400     box-shadow: 3px 3px 3px #808080 ;
2401 }
2402
2403
2404 .example {
2405     background-color: #f5f5eb ;
2406     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2407
2408 }
2409
2410 div.exampletitle{
2411     font-family: "URW Classico", Optima, "Linux Biolinum O",
2412         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2413         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2414     font-weight:bold;
2415     font-variant: small-caps ;
2416     border-bottom: 1px solid silver ;
2417     color: #304070 ;
2418     text-align: center ;
2419     text-shadow: 1px 1px 2px #808080;
2420 }
2421
2422
2423 .sidebar {
2424     background-color: #f5f5eb ;
2425     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2426
2427 }
2428
2429 div.sidebartitle{
2430     font-family: "URW Classico", Optima, "Linux Biolinum O",
2431         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2432         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2433     font-weight:bold;
```

```
2434     font-variant: small-caps ;
2435     border-bottom: 1px solid silver ;
2436     color: #304070 ;
2437     text-align: center ;
2438     text-shadow: 1px 1px 2px #808080;
2439 }
2440
2441
2442 .fancyvrblabel {
2443     font-family: "URW Classico", Optima, "Linux Biolinum O",
2444         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2445         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2446     font-weight: bold;
2447     font-variant: small-caps ;
2448     font-size: 1.5em ;
2449     color: #304070 ;
2450     text-align: center ;
2451     text-shadow: 1px 1px 2px #808080;
2452 }
2453
2454 div.minipage {
2455     background-color: #eeeeee7 ;
2456     border: 1px solid silver ;
2457     border-radius: 1ex ;
2458 }
2459
2460 div.framebox div.minipage {border:none ; background:none}
2461
2462 section.textbody > div.minipage {
2463     box-shadow: 3px 3px 3px #808080 ;
2464 }
2465
2466 div.fboxBlock div.minipage { box-shadow: none ; }
2467
2468 .framed .minipage , .framedleftbar .minipage {
2469     border: none ;
2470     background: none ;
2471     padding: 0ex ;
2472     margin: 0ex ;
2473 }
2474
2475 figure.figure .minipage, figcaption .minipage { border: none; }
2476
2477 div.marginblock div.minipage ,
2478 div.marginparblock div.minipage
2479     { border: none; }
2480
2481 figure , div.marginblock {
2482     background-color: #eeeeee7 ;
2483     border: 1px solid silver ;
```

```
2484     border-radius: 1ex ;
2485     box-shadow: 3px 3px 3px #808080 ;
2486 }
2487
2488 figure figure {
2489     border: 1px solid silver ;
2490     margin: 0em ;
2491     box-shadow: none ;
2492 }
2493
2494 /*
2495 figcaption {
2496     border-top: 1px solid silver ;
2497     border-bottom: 1px solid silver ;
2498     background-color: #e8e8e8 ;
2499 }
2500 */
2501
2502
2503 div.table {
2504     box-shadow: 3px 3px 3px #808080 ;
2505 }
2506
2507 /*
2508 .tnotes {
2509     background: #e8e8e8;
2510     border: 1px solid silver;
2511 }
2512 */
2513
2514
2515 nav.topnavigation{
2516     background-color: #b0b8b0 ;
2517     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2518 }
2519
2520 nav.botnavigation{
2521     background-color: #b0b8b0 ;
2522     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2523 }
2524
2525
2526
2527 header{
2528     background-color: #F7F7F0 ;
2529     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2530 }
2531
2532 footer{
2533     background-color: #F7F7F0 ;
```

```

2534     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2535 }
2536
2537
2538
2539 nav.sidetoc {
2540     background-color: #F7F7F0 ;
2541     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2542     box-shadow: 3px 3px 3px #808080 ;
2543     border-radius: 0px 0px 0px 20px ;
2544 }
2545
2546 div.sidetoc title {color: #304070 ; }
2547
2548 nav.sidetoc a: hover {
2549     color: #006000 ;
2550     text-decoration: none ;
2551     text-shadow: 0px 0px 2px #a0a0a0;
2552 }
2553
2554
2555 @media screen and (max-width: 45em) {
2556     nav.sidetoc { border-radius: 0 ; }
2557 }
2558
2559
2560 \end{filecontents*}
2561 % \end{Verbatim}% for syntax highlighting
2562 \end{warpprint}

```

34.6 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.

```

2563 \begin{warpprint}
2564 \begin{filecontents*}{lwarp_formal.css}
2565 @import url("lwarp.css") ;
2566
2567
2568
2569 A:link {color: #802020 ; text-decoration: none; }
2570 A:visited {color: #802020 ; text-shadow: none ;}
2571 A: hover {color: #400000 ; text-shadow: none ;}
2572 A: active {color: #C00000 ; text-shadow: none ;}

```

```
2573
2574
2575 body {
2576     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2577         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2578         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2579         "Times New Roman", serif;
2580     background: #fffcf5;
2581 }
2582
2583 span.textrm {
2584     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2585         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2586         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2587         "Times New Roman", serif;
2588 }
2589
2590 span.textsf {
2591     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2592         Geneva, Verdana, sans-serif ;
2593 }
2594
2595
2596
2597 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2598 {
2599     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2600         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2601         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2602         "Times New Roman", serif;
2603     color: #800000 ;
2604     text-shadow: none ;
2605 }
2606
2607 h1, h2 {
2608     background-color: #fffcf5 ;
2609     background-image: none ;
2610     border-bottom: 1px solid #808080;
2611     border-top: 2px solid #808080;
2612 }
2613
2614 div.abstracttitle {
2615     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2616         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2617         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2618         "Times New Roman", serif;
2619     color: black ;
2620     text-shadow: none ;
2621 }
2622
```

```
2623 span.abstractrunintitle {
2624     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2625         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2626         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2627         "Times New Roman", serif;
2628     color: black ;
2629     text-shadow: none ;
2630 }
2631
2632 div.abstract { font-size: 100% }
2633
2634 .sidebar {
2635     background: #fffcf5;
2636     background-image: none ;
2637     margin: 2em 5% 2em 5%;
2638     padding: 0.5em 1em;
2639     border: none ;
2640     border-top : 1px solid silver;
2641     border-bottom : 1px solid silver;
2642     font-size: 90% ;
2643 }
2644
2645 div.sidebartitle{
2646     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2647         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2648         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2649         "Times New Roman", serif;
2650     color: #800000 ;
2651     text-shadow: none ;
2652     border: none ;
2653 }
2654
2655 .example {
2656     background: #fffcf5;
2657     background-image: none ;
2658     margin: 2em 5% 2em 5%;
2659     padding: 0.5em 1em;
2660     border: none ;
2661     border-top : 1px solid silver;
2662     border-bottom : 1px solid silver;
2663 }
2664
2665 div.exampletitle{
2666     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2667         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2668         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2669         "Times New Roman", serif;
2670     color: #800000 ;
2671     text-shadow: none ;
2672     border: none ;
```

```
2673 }
2674
2675 div.fancyvrblabel{
2676     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2677         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2678         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2679         "Times New Roman", serif;
2680     color: #800000 ;
2681     text-shadow: none ;
2682     border: none ;
2683 }
2684
2685
2686
2687 .verse {
2688     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2689         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2690         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2691         "Times New Roman", serif;
2692 }
2693
2694
2695 figure {
2696     margin: 3ex 5% 3ex 5% ;
2697     padding: 1ex 1em 1ex 1em ;
2698     background-color: #fffcf5 ;
2699     overflow-x: auto ;
2700     border: none ;
2701 /*     border-top: 1px solid silver; */
2702 /*     border-bottom: 1px solid silver; */
2703 }
2704
2705
2706 figcaption , .lstlisting {
2707     border: none ;
2708 /*     border-top: 1px solid silver ; */
2709 /*     border-bottom: 1px solid silver ; */
2710     background-color: #fffcf5 ;
2711 }
2712
2713 .tnotes {
2714     background: #fffcf5 ;
2715 }
2716
2717 .theorem {
2718     background: none ;
2719 }
2720
2721 .minipage {
2722     background-color: #fffcf5 ;
```

```
2723     border: none ;
2724 }
2725
2726 div.floatrow figure { border: none ; }
2727
2728 figure figure { border: none ; }
2729
2730
2731 nav.toc, nav.lof, nav.lot, nav.lol {
2732     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2733         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2734         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2735         "Times New Roman", serif;
2736 }
2737
2738 nav.sidetoc {
2739     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2740         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2741         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2742         "Times New Roman", serif;
2743     background-image: linear-gradient(to bottom, #fffcf5, #COCOCO);
2744     border-radius: 0px 0px 0px 20px ;
2745 }
2746
2747 div.sidetoc{
2748     color: #800000 ;
2749 }
2750
2751 header{
2752     background-color: #e0e0e0 ;
2753     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2754     text-align:center ;
2755 }
2756
2757 footer{
2758     background-color: #e0e0e0 ;
2759     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
2760     padding: 2ex 1em 2ex 1em ;
2761     clear:right ;
2762     text-align:left ;
2763 }
2764
2765 nav.botnavigation {
2766     background: #dedcd5 ;
2767     border-top: 1px solid black ;
2768 }
2769 \end{filecontents*}
2770 % \end{Verbatim}% for syntax highlighting
2771 \end{warpprint}
```

34.7 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.

```

2772 \begin{warpprint}
2773 \begin{filecontents*}{sample_project.css}
2774 /* ( --- Start of project.css --- ) */
2775 /* ( --- A sample project-specific CSS file for lwarp --- ) */
2776
2777 /* Load default lwarp settings: */
2778 \import url("lwarp.css") ;
2779 /* or lwarp_formal.css, lwarp_sagebrush.css */
2780
2781 /* Project-specific CSS setting follow here. */
2782 /* . . . */
2783
2784 /* ( --- End of project.css --- ) */
2785 \end{filecontents*}
2786 % \end{Verbatim}% for syntax highlighting
2787 \end{warpprint}

```

34.8 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.

```

2788 \begin{warpprint}
2789 \begin{filecontents*}{lwarp.xdy}
2790 (require "tex/inputenc/latin.xdy")
2791 (merge-rule "\\PS *" "Postscript")
2792 (require "texindy.xdy")
2793 (require "page-ranges.xdy")
2794 (require "book-order.xdy")
2795 (require "page-ranges.xdy")
2796 (markup-locref :open "\hyperindexref{" :close "}")
2797 (define-location-class "arabic-page-numbers"
2798   ("arabic-numbers") :min-range-length 1)
2799 (define-location-class-order ("roman-page-numbers"
2800   "arabic-page-numbers"
2801   "alpha-page-numbers"
2802   "Roman-page-numbers"

```

```

2803             "Alpha-page-numbers"
2804             "see"
2805             "seealso"))
2806 \end{filecontents*}
2807 % \end{Verbatim}% for syntax highlighting
2808 \end{warpprint}

```

34.9 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.

```

2809 \begin{warpprint}
2810 \begin{filecontents*}{lwarp_mathjax.txt}
2811 <!-- https://groups.google.com/forum/#!topic/
2812             mathjax-users/jUtewUcE2bY -->
2813 <script type="text/x-mathjax-config">
2814 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2815     var seteqsectionDefault = {name: "", num: 0};
2816     var seteqsections = {}, seteqsection = seteqsectionDefault;
2817     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2818     var AMS = MathJax.Extension["TeX/AMSmath"];
2819     TEX.Definitions.Add({
2820     macros: {
2821         seteqsection: "mySection",
2822         seteqnumber: "mySetEqNumber"
2823     }
2824     });
2825
2826     PARSE.Augment({
2827     mySection: function (name) {
2828         seteqsection.num = AMS.number;
2829         var n = this.GetArgument(name);
2830         if (n === "") {
2831             seteqsection = seteqsectionDefault;
2832         } else {
2833             if (!seteqsections["_"+n])
2834                 seteqsections["_"+n] = {name:n, num:0};
2835             seteqsection = seteqsections["_"+n];
2836         }
2837         AMS.number = seteqsection.num;
2838     },
2839     mySetEqNumber: function (name) {
2840         var n = this.GetArgument(name);
2841         if (!n || !n.match(/^\s*[0-9]+\s$/))

```

```

2842         n = ""; else n = parseInt(n)-1;
2843         <!-- $ syntax highlighting -->
2844         if (n === "" || n < 1)
2845             TEX.Error
2846                 ("Argument to "+name+" should be a positive integer");
2847         AMS.number = n;
2848     }
2849 });
2850 MathJax.Hub.Config({
2851   TeX: {
2852     equationNumbers: {
2853       formatTag: function (n)
2854         {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
2855       formatID: function (n) {
2856         n = (seteqsection.name+'.'+n).replace
2857             (/[<>&]/g,"").replace(/\./,"");
2858         return 'mjax-eqn-' + n;
2859       }
2860     }
2861   }
2862 });
2863 });
2864 </script>
2865
2866 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2867 <script type="text/x-mathjax-config">
2868   MathJax.Ajax.config.path["Contrib"] =
2869     "https://cdn.mathjax.org/mathjax/contrib";
2870 </script>
2871
2872 <!-- https://github.com/burnpanck/MathJax-siunitx -->
2873
2874 <script type="text/x-mathjax-config">
2875   MathJax.Hub.Config({
2876     extensions: ["tex2jax.js","siunitx/siunitx.js"],
2877     jax: ["input/TeX","output/HTML-CSS"],
2878     tex2jax: {
2879       inlineMath: [["$","$"],["\\(","\\)"]] ,
2880       processClass: "tabbing|verse"
2881     },
2882     TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "sinuitx.js"]}
2883   });
2884   MathJax.Ajax.config.path['siunitx'] = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
2885 </script>
2886
2887 <script type="text/x-mathjax-config">
2888   MathJax.Hub.Config({
2889     TeX: {
2890       equationNumbers: {
2891         autoNumber: "AMS"

```

```

2892     }
2893   }
2894 });
2895 </script>
2896
2897 <!-- Alternative CDN provider: -->
2898 <script type="text/javascript" async
2899 src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
2900 </script>
2901
2902 <!-- No longer supported after April 30, 2017: -->
2903 <!--
2904 <script
2905   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
2906 </script>
2907 -->
2908
2909 \end{filecontents*}
2910 % \end{Verbatim}% for syntax highlighting
2911 \end{warpprint}

```

34.10 lwarpmk option

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

```

2912 \begin{LWR@createlwarpmk}

```

Prog lwarpmk Creates a local copy of lwarpmk:

```

2913 \begin{filecontents*}{lwarpmk.lua}
2914 #!/usr/bin/env texlua
2915
2916 -- Copyright 2016-2018 Brian Dunn
2917
2918 -- Print the usage of the lwarpmk command:
2919
2920 printversion = "v0.49"
2921
2922 function printhelp ()
2923 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2924 end
2925
2926 function printusage ()
2927 print ( [[
2928
2929 lwarpmk print [project]: Compile the print version if necessary.
2930 lwarpmk print1 [project]: Forced single compile of the print version.

```

```

2931 lwarpmk printindex [project]: Process the index for the print version.
2932 lwarpmk printglossary [project]: Process the glossary for the print version.
2933 lwarpmk html [project]: Compile the HTML version if necessary.
2934 lwarpmk html1 [project]: Forced single compile of the HTML version.
2935 lwarpmk htmlindex [project]: Process the index for the html version.
2936 lwarpmk htmlglossary [project]: Process the glossary for the html version.
2937 lwarpmk again [project]: Touch the source code to trigger recompiles.
2938 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2939 lwarpmk pdftohtml [project]:
2940     For use with latexmk or a Makefile:
2941     Convert project_html.pdf to project_html.html and
2942     individual HTML files.
2943 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
2944 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2945 lwarpmk -h: Print this help message.
2946 lwarpmk --help: Print this help message.
2947
2948 ]] )
2949 printconf ( )
2950 end
2951
2952 -- Print the format of the configuration file lwarpmk.conf:
2953
2954 function printconf ( )
2955 print ( [[
2956 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2957 --
2958 opsystem = "Unix"      (or "Windows")
2959 latexname = "pdflatex" (or "lualatex", or "xelatex")
2960 sourcename = "projectname" (the source-code filename w/o .tex)
2961 homehtmlfilename = "index" (or perhaps the project name)
2962 htmlfilename = ""      (or "projectname" - filename prefix)
2963 latexmk = "false"      (or "true" to use latexmk to build PDFs)
2964 language = "english"   (use a language supported by xindy)
2965 xdyfile = "lwarp.xdy"  (or a custom file based on lwarp.xdy)
2966 --
2967 Filenames must contain only letters, numbers, underscore, or dash.
2968 Values must be in "quotes".
2969
2970 ]] ) ;
2971 end
2972
2973
2974 -- Split one large sourcefile into a number of files,
2975 -- starting with destfile.
2976 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2977
2978 function splitfile (destfile,sourcefile)
2979 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2980 local sfile = io.open(sourcefile)

```

```

2981 io.output(destfile)
2982 for line in sfile:lines() do
2983 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*|") ;
2984 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2985 io.output(newfilename) ;
2986 else -- not a splitpoint
2987 io.write (line .. "\n") ;
2988 end
2989 end -- do
2990 io.close(sfile)
2991 end -- function
2992
2993 -- Incorrect value, so print an error and exit.
2994
2995 function cvalueerror ( line, linenum , cvalue )
2996     print ( linenum .. " : " .. line ) ;
2997     print ("lwarpmk: incorrect variable value \"" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2998     printconf () ;
2999     os.exit(1) ;
3000 end
3001
3002 -- Load settings from the project's "lwarpmk.conf" file:
3003
3004 function loadconf ()
3005 -- Default configuration filename:
3006 local conffile = "lwarpmk.conf"
3007 -- Optional configuration filename:
3008 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
3009 -- Default language:
3010 language = "english"
3011 -- Default xdyfile:
3012 xdyfile = "lwarp.xdy"
3013 -- Verify the file exists:
3014 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
3015 print("lwarpmk: " .. conffile .. " does not exist.")
3016 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
3017 printhelp () ;
3018 os.exit(1) -- exit the entire lwarpmk script
3019 else -- file exists
3020 -- Read the file:
3021 print ("lwarpmk: Reading " .. conffile .. ".")
3022 local cfile = io.open(conffile)
3023 -- Scan each line:
3024 local linenum = 0
3025 for line in cfile:lines() do -- scan lines
3026 linenum = linenum + 1
3027 i,j,cvarname,cvalue = string.find (line,"([%w-_*])%s*=%s*\"([%w%-_%.]*)\"") ;
3028 -- Error if incorrect enclosing characters:
3029 if ( i == nil ) then
3030 print ( linenum .. " : " .. line ) ;

```

```

3031 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
3032 printconf ( ) ;
3033 os.exit(1) ;
3034 end
3035 if ( cvarname == "opssystem" ) then
3036     -- Verify choice of opssystem:
3037     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
3038         opssystem = cvalue
3039     else
3040         cvalueerror ( line, linenum , cvalue )
3041     end
3042 elseif ( cvarname == "latexname" ) then
3043     -- Verify choice of LaTeX compiler:
3044     if (
3045         (cvalue == "pdflatex") or
3046         (cvalue == "xelatex") or
3047         (cvalue == "lualatex")
3048     ) then
3049         latexname = cvalue
3050     else
3051         cvalueerror ( line, linenum , cvalue )
3052     end
3053 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
3054 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
3055 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
3056 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
3057 elseif ( cvarname == "language" ) then language = cvalue
3058 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
3059 else
3060 print ( linenum .. " : " .. line ) ;
3061 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in " .. conffile .. ".\n" ) ;
3062 printconf ( ) ;
3063 os.exit(1) ;
3064 end
3065 end -- do scan lines
3066 io.close(cfile)
3067 end -- file exists
3068 -- Select some operating-system commands:
3069 if opssystem=="Unix" then -- For Unix / Linux / Mac OS:
3070     rmname = "rm"
3071     mvname = "mv"
3072     touchnamepre = "touch"
3073     touchnamepost = ""
3074     dirslash = "/"
3075     opquote= "\"'"
3076 elseif opssystem=="Windows" then -- For Windows
3077     rmname = "DEL"
3078     mvname = "MOVE"
3079     touchnamepre = "COPY /b"
3080     touchnamepost = "+,,"

```

```
3081 dirsslash = "\\\"
3082 opquote= "\"
3083 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
3084 end --- for Windows
3085
3086 -- set xindycmd according to pdflatex vs xelatex/lualatex:
3087 if ( latexname == "pdflatex" ) then
3088 xindycmd = "texindy -C utf8"
3089 glossarycmd = "xindy -C utf8"
3090 else
3091 xindycmd = "xindy -M texindy -C utf8"
3092 glossarycmd = "xindy -C utf8"
3093 end
3094
3095 end -- loadconf
3096
3097
3098 function refreshdate ()
3099 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
3100 end
3101
3102
3103 -- Scan the LaTeX log file for the phrase "Rerun to get",
3104 -- indicating that the file should be compiled again.
3105 -- Return true if found.
3106
3107 function reruntoget (filesorce)
3108 local fsource = io.open(filesorce)
3109 for line in fsource:lines() do
3110 if ( string.find(line,"Rerun to get") ~= nil ) then
3111     io.close(fsource)
3112     return true
3113 end
3114 end
3115 io.close(fsource)
3116 return false
3117 end
3118
3119
3120 -- Compile one time, return true if should compile again.
3121 -- fsuffix is "" for print, "_html" for HTML output.
3122
3123 function onetime (fsuffix)
3124 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
3125 err = os.execute(
3126 --     "echo " ..
3127     latexname .. " " .. sourcename..fsuffix )
3128 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
3129 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
3130 end
```

```
3131
3132
3133-- Compile up to five times.
3134-- fsuffix is "" for print, "_html" for HTML output
3135
3136function manytimes (fsuffix)
3137if onetime(fsuffix) == true then
3138if onetime(fsuffix) == true then
3139if onetime(fsuffix) == true then
3140if onetime(fsuffix) == true then
3141if onetime(fsuffix) == true then
3142end end end end end
3143end
3144
3145-- Exit if the given file does not exist.
3146
3147function verifyfileexists (filename)
3148if (lfs.attributes ( filename , "modification" ) == nil ) then
3149print ( "lwarpmk: " .. filename .. " not found." ) ;
3150os.exit (1) ;
3151end
3152end
3153
3154
3155-- Convert <project>_html.pdf into HTML files:
3156
3157function pdftohtml ()
3158    -- Convert to text:
3159    print ("lwarpmk: Converting " .. sourcename
3160        .. "_html.pdf to " .. sourcename .. "_html.html")
3161    os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
3162        .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
3163    -- Split the result into individual HTML files:
3164    splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
3165end
3166
3167
3168-- Remove auxiliary files:
3169
3170function removeaux ()
3171    os.execute ( rmname .. " " ..
3172        sourcename .. ".aux " .. sourcename .. "_html.aux " ..
3173        sourcename .. ".toc " .. sourcename .. "_html.toc " ..
3174        sourcename .. ".lof " .. sourcename .. "_html.lof " ..
3175        sourcename .. ".lot " .. sourcename .. "_html.lot " ..
3176        sourcename .. ".idx " .. sourcename .. "_html.idx " ..
3177        sourcename .. ".ind " .. sourcename .. "_html.ind " ..
3178        sourcename .. ".log " .. sourcename .. "_html.log " ..
3179        sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
3180        "*_html_inc.*"
```

```

3181         )
3182 end
3183
3184
3185
3186 -- Create lateximages based on lateximages.txt:
3187 function createlateximages ()
3188 print ("lwarpmk: Creating lateximages.")
3189 local limagesfile = io.open("lateximages.txt")
3190 -- Create the lateximages directory, ignore error if already exists
3191 err = os.execute("mkdir lateximages")
3192 -- Scan lateximages.txt
3193 for line in limagesfile:lines() do
3194 -- lwimgpage is the page number in the PDF which has the image
3195 -- lwimgnum is the sequential lateximage number to assign for the image
3196 i,j,lwimgpage,lwimgnum = string.find (line,"|(.*)|(.*)|")
3197 -- For each entry:
3198 if ( i~=nil ) then
3199 -- Separate out the image into its own single-page pdf:
3200 err = os.execute(
3201 "pdfseparate -f " .. lwimgpage .. " -l " ..
3202 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
3203 -- Crop the image:
3204 err = os.execute(
3205 "pdfcrop lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
3206 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3207 -- Convert the image to svg:
3208 err = os.execute(
3209 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
3210 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3211 -- Move the result into lateximages/:
3212 err = os.execute(
3213 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirsslash )
3214 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3215 -- Remove the temporary files:
3216 err = os.execute(
3217 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
3218 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
3219 end
3220 end -- do
3221 io.close(limagesfile)
3222 end -- function
3223
3224
3225 -- Use latexmk to compile source and index:
3226 -- fsuffix is "" for print, or "_html" for HTML
3227 function compilelatexmk ( fsuffix )
3228 -- The recorder option is required to detect changes in <project>.tex
3229 -- while we are loading <project>_html.tex.
3230 err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "

```

```

3231     .. "-e "
3232     .. opquote
3233     .. "$makeindex = q/" -- $
3234     .. xindycmd
3235     .. " -M " .. xdyfile
3236     .. " -L " .. language .. " /"
3237     .. opquote
3238     .. " -pdflatex=\"\" .. latexname .. " %O %S\" "
3239     .. sourcename..fsuffix ..".tex" ) ;
3240     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
3241 end
3242
3243
3244
3245 -- lwarpmk --version :
3246
3247 if (arg[1] == "--version") then
3248 print ( "lwarpmk: " .. printversion )
3249
3250 else -- not --version
3251
3252 -- print intro:
3253
3254 print ("lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package.")
3255
3256 -- lwarpmk print:
3257
3258 if arg[1] == "print" then
3259 loadconf ()
3260 if ( latexmk == "true" ) then
3261     compilelatexmk ("" )
3262     print ("lwarpmk: Done.")
3263 else -- not latexmk
3264     verifyfileexists (sourcename .. ".tex") ;
3265     -- See if up to date:
3266     if (
3267         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
3268         (
3269             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3270             lfs.attributes ( sourcename .. ".pdf" , "modification" )
3271         )
3272     ) then
3273         -- Recompile if not yet up to date:
3274         manytimes("")
3275         print ("lwarpmk: Done." ) ;
3276     else
3277         print ("lwarpmk: " .. sourcename .. ".pdf is up to date." ) ;
3278     end
3279 end -- not latexmk
3280

```

```
3281 elseif arg[1] == "print1" then
3282     loadconf ()
3283     verifyfileexists (sourcename .. ".tex") ;
3284     onetime("")
3285     print ("lwarpmk: Done.") ;
3286
3287 -- lwarp printindex:
3288 -- Compile the index then touch the source
3289 -- to trigger a recompile of the document:
3290
3291 elseif arg[1] == "printindex" then
3292 loadconf ()
3293 print ("lwarpmk: Processing the index.")
3294 os.execute(
3295     xindycmd
3296     .. " -M " .. xdyfile
3297     .. " -L " .. language
3298     .. " " .. sourcename .. ".idx")
3299 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3300 refreshdate ()
3301 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3302 print ("lwarpmk: Done.")
3303
3304 -- lwarp printglossary:
3305 -- Compile the glossary then touch the source
3306 -- to trigger a recompile of the document:
3307
3308 elseif arg[1] == "printglossary" then
3309 loadconf ()
3310 print ("lwarpmk: Processing the glossary.")
3311
3312 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3313     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
3314     .. sourcename .. ".glo")
3315 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3316 refreshdate ()
3317 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3318 print ("lwarpmk: Done.")
3319
3320 -- lwarpmk html:
3321
3322 elseif arg[1] == "html" then
3323 loadconf ()
3324 if ( latexmk == "true" ) then
3325     compilelatexmk ("_html")
3326     pdftohtml ()
3327     print ("lwarpmk: Done.")
3328 else -- not latexmk
3329     verifyfileexists ( sourcename .. ".tex" ) ;
3330     -- See if exists and is up to date:
```

```
3331     if (
3332         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3333         (
3334             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3335             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3336         )
3337     ) then
3338         -- Recompile if not yet up to date:
3339         manytimes("_html")
3340         pdftohtml ()
3341         print ("lwarpmk: Done.")
3342     else
3343         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3344     end
3345 end -- not latexmk
3346
3347 elseif arg[1] == "html1" then
3348     loadconf ()
3349     verifyfileexists ( sourcename .. ".tex" ) ;
3350     onetime("_html")
3351     pdftohtml ()
3352     print ("lwarpmk: Done.")
3353
3354 elseif arg[1] == "pdftohtml" then
3355     loadconf ()
3356     pdftohtml ()
3357
3358 -- lwarpmk htmlindex:
3359 -- Compile the index then touch the source
3360 -- to trigger a recompile of the document:
3361
3362 elseif arg[1] == "htmlindex" then
3363     loadconf ()
3364     print ("lwarpmk: Processing the index.")
3365     os.execute(
3366         xindycmd
3367         .. " -M " .. xdyfile
3368         .. " -L " .. language
3369         .. " " .. sourcename .. "_html.idx"
3370     )
3371     print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3372     refreshdate ()
3373     print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3374     print ("lwarpmk: Done.")
3375
3376 -- lwarpmk htmlglossary:
3377 -- Compile the glossary then touch the source
3378 -- to trigger a recompile of the document:
3379
3380 elseif arg[1] == "htmlglossary" then
```

```
3381 loadconf ()
3382 print ("lwarpmk: Processing the glossary.")
3383
3384 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3385     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3386     "_html.gls " .. sourcename .. "_html.glo")
3387
3388 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3389 refreshdate ()
3390 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3391 print ("lwarpmk: Done.")
3392
3393 -- lwarpmk limages:
3394 -- Scan the lateximages.txt file to create lateximages,
3395 -- then touch the source to trigger a recompile.
3396
3397 elseif arg[1] == "limages" then
3398 loadconf ()
3399 print ("lwarpmk: Processing images.")
3400 createlateximages ()
3401 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3402 refreshdate ()
3403 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3404 print ("lwarpmk: Done.")
3405
3406 -- lwarpmk again:
3407 -- Touch the source to trigger a recompile.
3408
3409 elseif arg[1] == "again" then
3410 loadconf ()
3411 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3412 refreshdate ()
3413 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3414 print ("lwarpmk: Done.")
3415
3416 -- lwarpmk clean:
3417 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3418
3419 elseif arg[1] == "clean" then
3420 loadconf ()
3421 removeaux ()
3422 print ("lwarpmk: Done.")
3423
3424 -- lwarpmk cleanall
3425 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3426 -- and also project.pdf, *.html
3427
3428 elseif arg[1] == "cleanall" then
3429 loadconf ()
3430 removeaux ()
```

```

3431 os.execute ( rmname .. " " ..
3432     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
3433     "*.html"
3434 )
3435 print ("lwarpmk: Done.")
3436
3437 -- lwarpmk with no argument :
3438
3439 elseif (arg[1] == nil) then
3440 printhelp ()
3441
3442 -- lwarpmk -h or lwarpmk --help :
3443
3444 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3445 printusage ()
3446
3447 else
3448 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
3449 printhelp ()
3450 end
3451
3452 end -- not --version
3453 \end{filecontents*}
3454 % \end{Verbatim}% for syntax highlighting

3455 \end{LWR@createlwarpmk}

```

35 Stacks

for HTML output: 3456 \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.

35.1 Assigning depths

initial depths for empty stack entries:

```
3457 \newcommand*{\LWR@depthnone}{-5}
```

all sectioning depths are deeper than LWR@depthfinished:

```
3458 \newcommand*{\LWR@depthfinished}{-4}
3459 \newcommand*{\LWR@depthpart}{-1}
3460 \newcommand*{\LWR@depthchapter}{0}
3461 \newcommand*{\LWR@depthsection}{1}
3462 \newcommand*{\LWR@depthsubsection}{2}
3463 \newcommand*{\LWR@depthsubsubsection}{3}
3464 \newcommand*{\LWR@depthparagraph}{4}
3465 \newcommand*{\LWR@depthsubparagraph}{5}
```

used by \itemize, \enumerate, \description:

```
3466 \newcommand*{\LWR@depthlist}{6}
```

used by \item:

```
3467 \newcommand*{\LWR@depthlistitem}{7}
```

35.2 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.

```
3468 \newcommand*{\LWR@closeone}{}% top of the stack
3469 \newcommand*{\LWR@closetwo}{}
3470 \newcommand*{\LWR@closethree}{}
3471 \newcommand*{\LWR@closefour}{}
3472 \newcommand*{\LWR@closefive}{}
3473 \newcommand*{\LWR@closesix}{}
3474 \newcommand*{\LWR@closeseven}{}
3475 \newcommand*{\LWR@closeeight}{}
3476 \newcommand*{\LWR@closenine}{}
3477 \newcommand*{\LWR@closeten}{}
3478 \newcommand*{\LWR@closeeleven}{}
3479 \newcommand*{\LWR@closetwelve}{}%
```

35.3 Closing depths

A stack to record the depth of each level:



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

Ex:

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

```
3480 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
3481 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
3482 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
3483 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
3484 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
3485 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
3486 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
3487 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
3488 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
3489 \newcommand*{\LWR@closedephten}{\LWR@depthnone}
3490 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3491 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

35.4 Pushing and popping the stack

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

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

```
3492 \NewDocumentCommand{\pushclose}{m m}
3493 {
3494 \global\let\LWR@closetwelve\LWR@closeeleven
3495 \global\let\LWR@closeeleven\LWR@closeten
3496 \global\let\LWR@closeten\LWR@closenine
3497 \global\let\LWR@closenine\LWR@closeeight
3498 \global\let\LWR@closeeight\LWR@closeseven
3499 \global\let\LWR@closeseven\LWR@closesix
3500 \global\let\LWR@closesix\LWR@closefive
```

```

3501 \global\let\LWR@closefive\LWR@closefour
3502 \global\let\LWR@closefour\LWR@closethree
3503 \global\let\LWR@closethree\LWR@closetwo
3504 \global\let\LWR@closetwo\LWR@closeone
3505 \global\let\LWR@closeone#1
3506 \global\let\LWR@closedepthtwelve\LWR@closedeptheleven
3507 \global\let\LWR@closedeptheleven\LWR@closedephten
3508 \global\let\LWR@closedephten\LWR@closedepthnine
3509 \global\let\LWR@closedepthnine\LWR@closedeptheight
3510 \global\let\LWR@closedeptheight\LWR@closedepthseven
3511 \global\let\LWR@closedepthseven\LWR@closedepthsix
3512 \global\let\LWR@closedepthsix\LWR@closedepthfive
3513 \global\let\LWR@closedepthfive\LWR@closedepthfour
3514 \global\let\LWR@closedepthfour\LWR@closedepththree
3515 \global\let\LWR@closedepththree\LWR@closedepthtwo
3516 \global\let\LWR@closedepthtwo\LWR@closedepthone
3517 \global\let\LWR@closedepthone#2
3518 }

```

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

```

3519 \newcommand*{\popclose}
3520 {
3521 \global\let\LWR@closeone\LWR@closetwo
3522 \global\let\LWR@closetwo\LWR@closethree
3523 \global\let\LWR@closethree\LWR@closefour
3524 \global\let\LWR@closefour\LWR@closefive
3525 \global\let\LWR@closefive\LWR@closesix
3526 \global\let\LWR@closesix\LWR@closeseven
3527 \global\let\LWR@closeseven\LWR@closeeight
3528 \global\let\LWR@closeeight\LWR@closenine
3529 \global\let\LWR@closenine\LWR@closeten
3530 \global\let\LWR@closeten\LWR@closeeleven
3531 \global\let\LWR@closeeleven\LWR@closetwelve
3532 \global\let\LWR@closedepthone\LWR@closedepthtwo
3533 \global\let\LWR@closedepthtwo\LWR@closedepththree
3534 \global\let\LWR@closedepththree\LWR@closedepthfour
3535 \global\let\LWR@closedepthfour\LWR@closedepthfive
3536 \global\let\LWR@closedepthfive\LWR@closedepthsix
3537 \global\let\LWR@closedepthsix\LWR@closedepthseven
3538 \global\let\LWR@closedepthseven\LWR@closedeptheight
3539 \global\let\LWR@closedeptheight\LWR@closedepthnine
3540 \global\let\LWR@closedepthnine\LWR@closedephten
3541 \global\let\LWR@closedephten\LWR@closedeptheleven
3542 \global\let\LWR@closedeptheleven\LWR@closedepthtwelve
3543 }

3544 \end{warpHTML}

```

36 Data arrays

These macros are similar to the arrayjobx package, except that \LWR@setexparray's argument is expanded only once when assigned.

name has no backslash, index can be a number or a text name, and an empty value must be \relax instead of empty.

To assign an empty value:

```
\LWR@setexparray{name}{index}{}
```

for HTML output: 3545 \begin{warpHTML}

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

```
3546 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
3547 \ifstrempy{#3}%
3548 {\csdef{#1#2}{}}%
3549 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3550 }
```

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

```
3551 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}
```

```
3552 \end{warpHTML}
```

37 Sanitizing labels and filenames

Special handling for underscores in labels and filenames.

for HTML output: 3553 \begin{warpHTML}

\LWR@sanitized The sanitized version of what was given to \LWR@sanitize. Characters are set to their detokenized versions. Required for underscores in labels and filenames.

```
3554 \newcommand*{\LWR@sanitized}{}%
```

```
\LWR@sanitize {<text>}
```

Sanitizes the text and returns the result in \LWR@sanitized.

```

3555 \newcommand*{\LWR@sanitize}[1]{%
3556 \LWR@traceinfo{\LWR@sanitize: !#1!}%
3557 \edef\LWR@sanitized{#1}%
3558 \LWR@traceinfo{\LWR@sanitize expanded: !\LWR@sanitized!}%
3559 \edef\LWR@sanitized{\detokenize\expandafter{\LWR@sanitized}}%
3560 \LWR@traceinfo{\LWR@sanitize result: !\LWR@sanitized!}%
3561 }

3562 \end{warpHTML}

```

38 HTML entities

for HTML output: 3563 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
3564 \let\LWR@origampersand\&
```

\HTMLentity {<entitytag>}

```

3565 \newcommand*{\HTMLentity}[1]{%
3566 % \LWR@traceinfo{\HTMLentity \detokenize{#1}}%
3567 \begingroup%
3568 \LWR@FBcancel%
3569 \LWR@origampersand#1;%
3570 \endgroup
3571 % \LWR@traceinfo{\HTMLentity done}%
3572 }

```

\HTMLunicode {<hex_unicode>}

```
3573 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\#x#1}}
```

\&

```
3574 \renewrobustcmd*{\&}{\HTMLentity{amp}}
```

\textless

\textgreater

```

3575 \let\LWR@origtextless\textless
3576 \renewcommand*{\textless}{\HTMLentity{lt}}
3577
3578 \let\LWR@origtextgreater\textgreater
3579 \renewcommand*{\textgreater}{\HTMLentity{gt}}

```

```
3580 \end{warpHTML}
```

39 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: 3581 `\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.

```
3582 \providecommand*{\BaseJobname}{\jobname}
```

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

```
3583 \providecommand*{\HTMLFilename}{}
```

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

```
3584 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

`\SetHTMLFileNumber` `{<number>}`

Sets the file number for the next file to be generated. 0 is the home page. Use just before the next sectioning command, and set it to one less than the desired number of the next section. May be used to generate numbered groups of nodes such as 100+ for one chapter, 200+ for another chapter, etc.

```
3585 \newcommand*{\SetHTMLFileNumber}[1]{%
3586 \setcounter{LWR@htmlfilenumber}{#1}%
3587 }
```

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

Defaults to use section names in the filenames.

```
3588 \newbool{FileSectionNames}
3589 \booltrue{FileSectionNames}
```

```
3590 \end{warpall}
```

for HTML output: 3591 \begin{warpHTML}

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

```
3592 \newcounter{LWR@htmlfilenumber}
3593 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
3594 \newcommand*{\LWR@htmlsectionfilename}[1]{%
3595 \LWR@traceinfo{LWR@htmlsectionfilename A !\detokenize{#1}!}%
```

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

```
3596 % \LWR@traceinfo{about to assign temp}%
3597 \edef\LWR@tempone{#1}%
3598 \LWR@traceinfo{about to compare with ??}%
3599 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3600 {\LWR@traceinfo{found ??}}%
3601 {\LWR@traceinfo{not found ??}}%
3602 \LWR@traceinfo{about to compare with zero or empty}%
3603 \ifthenelse{%
3604   \equal{\LWR@tempone}{0}%
3605   \OR \equal{\LWR@tempone}{}%
3606   \OR \equal{\LWR@tempone}{??}}%
3607 }%
3608 {%
3609   \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3610   \HomeHTMLFilename.html%
3611 }%
```

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:

```
3612 {%
3613   \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
3614   \ifthenelse{%
3615     \equal{\HTMLFilename}{ } \AND
3616     \equal{\LWR@tempone}{Index} \OR
3617     \equal{\LWR@tempone}{index}%
3618   }%
3619   {%
```

```

3620      \LWR@traceinfo{Prefixing the index name with an underscore.}%
3621      \_#1.html%
3622  }%

```

Otherwise, create a filename with the chosen prefix:

```

3623      {\HTMLFilename#1.html}%
3624 }%
3625 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
3626 }

```

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

Prints the filename for the given label

```

3627 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3628 \LWR@traceinfo{LWR@htmlrefsectionfilename: !\detokenize{#1}!}%
3629 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
3630 \LWR@traceinfo{LWR@htmlrefsectionfilename: done}%
3631 }

3632 \end{warpHTML}

```

40 Homepage link

for HTML output: 3633 \begin{warpHTML}

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

```

3634 \newcommand*{\LinkHome}{%
3635 \LWR@subhyperrefclass{%
3636 \HomeHTMLFilename.html}%
3637 {Home}{linkhome}%
3638 }

```

`\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.

```

3639 \newcommand*{\LWR@topnavigation}{
3640 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
3641 }

```

`\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.

```
3642 \newcommand*{\LWR@botnavigation}{
3643 \LWR@htmlElementclassline{nav}{botnavigation}{\LinkHome}
3644 }

3645 \end{warpHTML}
```

41 \LWRPrintStack diagnostic tool



Diagnostics tool: Prints the \LaTeX nesting depth values for the stack levels. `\LWR@startpars` is used before printing the stack, so that `\LWRPrintStack` may be called from anywhere in the normal text flow.

for HTML output: 3646 `\begin{warpHTML}`

`\LWRPrintStack` Prints the closedepth stack.

```
3647 \newcommand*{\LWR@subprintstack}{
3648 \LWR@closedepthone\ \LWR@closedepthtwo\ \LWR@closedepththree\
3649 \LWR@closedepthfour\ \LWR@closedepthfive\ \LWR@closedepthsix\
3650 \LWR@closedepthseven\ \LWR@closedeptheight\ \LWR@closedepthnine\
3651 \LWR@closedephten\ \LWR@closedeptheleven\ \LWR@closedephtwelve\
3652 }
3653
3654 \newcommand*{\LWRPrintStack}{
3655 \LWR@startpars
3656 \LWR@subprintstack
3657 }

3658 \end{warpHTML}
```

for PRINT output: 3659 `\begin{warpprint}`

```
3660 \newcommand*{\LWRPrintStack}{}

3661 \end{warpprint}
```

42 Closing stack levels

for HTML output: 3662 `\begin{warpHTML}`

Close one nested level:

```
3663 \newcommand*{\LWR@closeoneprevious}{%
3664
3665 \LWR@closeone
3666
3667 \popclose
3668 }
```

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

```
3669 \newcommand*{\LWR@closeprevious}[1]{
3670 \LWR@traceinfo{\LWR@closeprevious to depth #1, depths are \LWR@subprintstack}%
```

Close any pending paragraph:

```
3671 \LWR@stoppars%
```

Close anything nested deeper than the desired depth. First close anything deeper, then at most one of the same level.

```
3672 \whilebool{test{\ifnumcomp{\LWR@closedepthone}{>}{#1}}}%
3673 {%
3674   \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
3675   \LWR@closeoneprevious%
3676 }%
3677 \ifbool{test{\ifnumcomp{\LWR@closedepthone}{=}{#1}}}%
3678 {%
3679   \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
3680   \LWR@closeoneprevious%
3681 }}%
3682 \LWR@traceinfo{\LWR@closeprevious: done, depths are \LWR@subprintstack}%
3683 }

3684 \end{warpHTML}
```

43 PDF pages and styles

for HTML output: 3685 `\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.

```
3686 \newcommand{\LWR@forcenewpage}{%  
3687 \ifinner\else%  
3688 \LWR@stoppars\LWR@orignewpage\LWR@startpars%  
3689 \fi%  
3690 }
```

\pagestyle, etc. are nullified for HTML output.

\pagestyle {\langle style\rangle}

```
3691 \renewcommand*{\pagestyle}[1]{}
```

\thispagestyle {\langle style\rangle}

```
3692 \renewcommand*{\thispagestyle}[1]{}
```

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

```
3693 \renewcommand*{\markboth}[2]{}
```

\markright {\langle right\rangle}

```
3694 \renewcommand*{\markright}[1]{}
```

\raggedbottom

```
3695 \renewcommand*{\raggedbottom}{} 
```

\flushbottom

```
3696 \renewcommand*{\flushbottom}{} 
```

\sloppy

```
3697 \renewcommand*{\sloppy}{} 
```

\fussy

```
3698 \renewcommand*{\fussy}{} 
```

\pagenumbering * {\langle commands\rangle}

```
3699 \RenewDocumentCommand{\pagenumbering}{s m}{} 
```

```
3700 \end{warpHTML}
```

44 HTML tags, spans, divs, elements

for HTML output: 3701 \begin{warpHTML}

44.1 Mapping \TeX Sections to HTML Sections

```
3702 \newcommand*{\LWR@tagtitle}{h1}
3703 \newcommand*{\LWR@tagtitleend}{/h1}
3704 \newcommand*{\LWR@tagpart}{h2}
3705 \newcommand*{\LWR@tagpartend}{/h2}
3706 \newcommand*{\LWR@tagchapter}{h3}
3707 \newcommand*{\LWR@tagchapterend}{/h3}
3708 \newcommand*{\LWR@tagsection}{h4}
3709 \newcommand*{\LWR@tagsectionend}{/h4}
3710 \newcommand*{\LWR@tagsubsection}{h5}
3711 \newcommand*{\LWR@tagsubsectionend}{/h5}
3712 \newcommand*{\LWR@tagsubsubsection}{h6}
3713 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
3714 \newcommand*{\LWR@tagparagraph}{}
3715 \newcommand*{\LWR@tagparagraphend}{/span}
3716 \newcommand*{\LWR@tagsubparagraph}{}
3717 \newcommand*{\LWR@tagsubparagraphend}{/span}
3718
3719 \newcommand*{\LWR@tagregularparagraph}{p}
```

44.2 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.)

```
3720 \providecommand*{\LWR@FBcancel}{}
3721
3722 \AtBeginDocument{%
3723 \ifundefined{frenchbsetup}%
3724 {}%
3725 {%
3726     \frenchbsetup{FrenchFootnotes=false}%
3727 %
3728     \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
```

```

3729 \renewrobustcmd*{\FBcolonspace}{%
3730     \begingroup%
3731     \LWR@FBcancel%
3732     \LWR@origampersand{} \nbsp;%
3733     \endgroup%
3734 }%
3735 \renewrobustcmd*{\FBthinspace}{%
3736     \begingroup%
3737     \LWR@FBcancel%
3738     \LWR@origampersand{\#x202f;% \,
3739     \endgroup%
3740 }%
3741 \renewrobustcmd*{\FBguillspace}{%
3742     \begingroup%
3743     \LWR@FBcancel%
3744     \LWR@origampersand{} \nbsp;% ~, for \og xyz \fg{}
3745     \endgroup%
3746 }%
3747 \DeclareDocumentCommand{\FBmedkern}{-}{%
3748     \begingroup%
3749     \LWR@FBcancel%
3750     \LWR@origampersand{\#x202f;% \,
3751     \endgroup%
3752 }%
3753 \DeclareDocumentCommand{\FBthickkern}{-}{%
3754     \begingroup%
3755     \LWR@FBcancel%
3756     \LWR@origampersand{} \nbsp;% ~
3757     \endgroup%
3758 }%
3759 \renewrobustcmd*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
3760 \ifFBunicode%
3761 \else%
3762     \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
3763     \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
3764 \fi%
3765 }%
3766 }

```

44.3 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.

```

3767 \newcommand*{\LWR@htmltagc}[1]{%
3768 {%

```

```

3769 \LWR@traceinfo{LWR@htmltagc !\detokenize{#1}!}%
3770 \begingroup%
3771 \LWR@FBcancel%
3772 \ifmode\else\protect\LWR@origttfamily\fi%
3773 \protect\LWR@origtextless%
3774 #1%
3775 \protect\LWR@origtextgreater%
3776 \endgroup%
3777 }%
3778 }

```

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.

```

3779 \newcommand*{\LWR@nestspanitem}{%
3780 \ifnewlist\else{\LWR@htmltagc{br /}}\fi%
3781 \LWR@origitem%
3782 }
3783
3784 \newenvironment*{LWR@nestspan}
3785 {%
3786 \LWR@traceinfo{LWR@nestspan starting}%
3787 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
3788 {%
3789   \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3790 }%
3791 {% not in a lateximage
3792   \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3793   \addtocounter{LWR@spandepth}{1}%
3794   \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
3795   \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3796   \renewcommand{\BlockClassSingle}[2]{##2}%
3797   \renewcommand{\LWR@forcenewpage}{-}%
3798   \renewcommand{\LWR@liststart}{%
3799     \let\item\LWR@nestspanitem%
3800   }%
3801   \renewcommand{\LWR@listend}{\LWR@htmltagc{br /}\LWR@htmltagc{br /}}%
3802 }% not in a lateximage
3803 \LWR@traceinfo{LWR@nestspan starting: done}%
3804 }% starting env
3805 {% ending env
3806 \LWR@traceinfo{LWR@nestspan ending}%
3807 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
3808 {}%
3809 {\addtocounter{LWR@spandepth}{-1}}%

```

```

3810 \LWR@traceinfo{LWR@nestspan ending: done}%
3811 }
3812
3813 \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.

```

3814 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3815 \LWR@ensuredoingapar%
3816 \LWR@htmltagc{#1}%
3817 \begin{LWR@nestspan}%
3818 #2%
3819 \LWR@htmltagc{/#1}%
3820 \end{LWR@nestspan}%
3821 }

```

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

```

3822 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3823 \LWR@traceinfo{LWR@htmlspanclass #2}%
3824 \LWR@ensuredoingapar%
3825 \LWR@subhtmlmlelementclass{span}[#1]{#2}%
3826 \begin{LWR@nestspan}%
3827 #3%
3828 \LWR@htmltagc{/span}%
3829 \LWR@traceinfo{LWR@htmlspanclass done}%
3830 \end{LWR@nestspan}%
3831 }

```

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

Print an HTML tag: `<tag>`

```

3832 \newcommand*{\LWR@htmltag}[1]{%
3833 % \LWR@traceinfo{LWR@htmltagb !\detokenize{#1}!}%
3834 \LWR@htmltagc{#1}%
3835 % \LWR@traceinfo{LWR@htmltagb: done}%
3836 }

```

44.4 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment
3837 \newcommand*{\LWR@htmlopencomment}{%
3838 {%
3839 % \LWR@traceinfo{\LWR@htmlopencomment}%
3840 \begingroup%
3841 \LWR@FBcancel%
3842 \ifmmode\else\protect\LWR@origttfamily\fi%
3843 \LWR@origmbox{\LWR@origtextless{!}{-}{-}}%
3844 \endgroup%
3845 }%
3846 }
3847
3848 \newcommand*{\LWR@htmlclosecomment}{%
3849 {%
3850 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3851 \begingroup%
3852 \LWR@FBcancel%
3853 \ifmmode\else\protect\LWR@origttfamily\fi%
3854 \LWR@origmbox{{-}{-}\LWR@origtextgreater}%
3855 \endgroup%
3856 }%
3857 }

\LWR@htmlcomment  {\langle comment \rangle}

3858 \newcommand{\LWR@htmlcomment}[1]{%
3859 \LWR@htmlopencomment}%
3860 {%
3861 \LWR@origttfamily% break ligatures
3862 #1%
3863 }%
3864 \LWR@htmlclosecomment{}}

\LWR@htmlblockcomment  {\langle comment \rangle}

3865 \newcommand{\LWR@htmlblockcomment}[1]
3866 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars}

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

```

```

3867 \newcommand*{\LWR@htmlblocktag}[1]{%
3868 \LWR@stoppars%
3869 \LWR@htmltag{#1}%
3870 \LWR@startpars%
3871 }

```

44.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```

3872 \NewDocumentCommand{\LWR@subhtmlclass}{m O{} m}{%
3873 \LWR@traceinfo{\LWR@subhtmlclass #1 #3}%
3874 \ifblank{#2}%
3875 {\LWR@htmltag{#1 class="#3"}}% empty option
3876 {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
3877 \LWR@traceinfo{\LWR@subhtmlclass done}%
3878 }

```

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

```

3879 \NewDocumentCommand{\LWR@htmlclass}{m o m}{%
3880 \LWR@stoppars%
3881 \LWR@subhtmlclass{#1}[#2]{#3}%
3882 \LWR@startpars%
3883 }

```

```
\LWR@htmlclassend {<element>} {<class>}
```

```

3884 \newcommand*{\LWR@htmlclassend}[2]{%
3885 \LWR@stoppars%
3886 \LWR@htmltag{/#1}%
3887 \ifbool{HTMLDebugComments}{%
3888   \LWR@htmlcomment{End of #1 ‘‘#2’’}%
3889 }{}%
3890 \LWR@startpars%
3891 }

```

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

```
3892 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
```

```
3893 \LWR@htmlelementclass{div}[#1]{#2}%
3894 }
```

```
\LWR@htmldivclassend {<class>}
```

```
3898 \newcommand*\LWR@htmldivclassend[1]{%
3896 \LWR@htmlelementclassend{div}{#1}%
3897 }
```

44.6 Single-line elements

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

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

```
3898 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
3899 \LWR@stoppars
3900 \LWR@subhtmlelementclass{#1}[#2]{#3}%
3901 #4%
3902 \LWR@htmltag{/#1}
3903 \LWR@startpars
3904 }
```

44.7 HTML5 semantic elements

```
\LWR@htmlelement {<element>}
```

```
3905 \newcommand*\LWR@htmlelement[1]{%
3906 \LWR@htmlblocktag{#1}
3907 }
```

```
\LWR@htmlelementend {<element>}
```

```
3908 \newcommand*\LWR@htmlelementend[1]{%
3909 \LWR@stoppars
3910 \LWR@htmltag{/#1}
3911 \LWR@startpars
3912 }
3913
3914 \end{warpHTML}
```

44.8 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 80.

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

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

for HTML output: 3915 `\begin{warpHTML}`
 3916 `\NewDocumentEnvironment{BlockClass}{o m}%`
 3917 `{\LWR@htmldivclass[#1]{#2}}`
 3918 `{\LWR@htmldivclassend{#2}}`
 3919 `\end{warpHTML}`

for PRINT output: 3920 `\begin{warpprint}`
 3921 `\NewDocumentEnvironment{BlockClass}{o m}{-}{-}%`
 3922 `\end{warpprint}`

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

for HTML output: 3923 `\begin{warpHTML}`
 3924 `\newcommand{\BlockClassSingle}[2]{%`
 3925 `\LWR@html@elementclassline{div}{#1}{#2}%`
 3926 `}`
 3927 `\end{warpHTML}`

for PRINT output: 3928 `\begin{warpprint}`
 3929 `\newcommand{\BlockClassSingle}[2]{#2}`
 3930 `\end{warpprint}`

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

for HTML output: 3931 `\begin{warpHTML}`
 3932 `\NewDocumentCommand{\InlineClass}{o m +m}{%`
 3933 `\LWR@htmlspanclass[#1]{#2}{#3}%`
 3934 `}`
 3935 `\end{warpHTML}`

for PRINT output: 3936 `\begin{warpprint}`
 3937 `\NewDocumentCommand{\InlineClass}{o m +m}{#3}%`
 3938 `\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:

```

3939 \begin{warpHTML}
3940 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{%
3941 {%
3942 \LWR@stoppars%
3943 \ifbool{FormatWP}%
3944 {%
3945     \addtocounter{LWR@thisautoidWP}{1}%
3946     \LWR@htmltag{%
3947         div class="#3" %
3948         id="\LWR@origmbox{autoidWP-\arabic{LWR@thisautoidWP}}"%
3949         \ifblank{#1}{\{ style="#1"%
3950     }%
3951 }% FormatWP
3952 {% not FormatWP
3953     \LWR@htmltag{%
3954         div class="#3"%
3955         \ifblank{#2}{\{ style="#2"%
3956     }%
3957 }% not FormatWP
3958 \LWR@startpars%
3959 }
3960 {\LWR@htmldivclassend{#3}}
3961 \end{warpHTML}

```

for PRINT output:

```

3962 \begin{warpprint}
3963 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{\{}}{%
3964 \end{warpprint}

```

44.9 Closing HTML tags

for HTML output: 3965 `\begin{warpHTML}`

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

```

3966 \newcommand*{\LWR@printclosepart}
3967     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3968 \newcommand*{\LWR@printclosechapter}
3969     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3970 \newcommand*{\LWR@printclosesection}
3971     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3972 \newcommand*{\LWR@printclosesubsection}
3973     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3974 \newcommand*{\LWR@printclosesubsubsection}

```

```

3975    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3976 \newcommand*{\LWR@printcloseparagraph}
3977    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3978 \newcommand*{\LWR@printclosesubparagraph}
3979    {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

3980 \newcommand*{\LWR@printcloselistitem}
3981    {\LWR@htmltag{/li}}
3982 \newcommand*{\LWR@printclosedescitem}
3983    {\LWR@htmltag{/dd}}
3984 \newcommand*{\LWR@printcloseitemize}
3985    {\LWR@htmltag{/ul}}
3986 \newcommand*{\LWR@printcloseenumerate}
3987    {\LWR@htmltag{/ol}}
3988 \newcommand*{\LWR@printclosedescription}
3989    {\LWR@htmltag{/dl}}

3990 \end{warpHTML}

```

45 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 46 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: 3991 `\begin{warpHTML}`

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

```

3992 \newcounter{LWR@spandepth}
3993 \setcounter{LWR@spandepth}{0}

```

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

```

3994 \newbool{LWR@doingstartpars}
3995 \boolfalse{LWR@doingstartpars}

```

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

```
3996 \newbool{LWR@doingapar}
3997 \global\boolfalse{LWR@doingapar}
```

\LWR@ensuredoiningapar 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:

```
3998 \newcommand*{\LWR@ensuredoiningapar}{%
3999 \ifbool{LWR@doingstartpars}%
4000 {\global\booltrue{LWR@doingapar}}%
4001 {}%
4002 }
```

\LWR@openparagraph

```
4003 \newcommand*{\LWR@openparagraph}
4004 {%
```

See if paragraph handling is enabled:

```
4005 \ifbool{LWR@doingstartpars}%
4006 {% handling pars
```

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

```
4007 \ifbool{expr{
4008 test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
4009 test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
4010 }% 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 .

```
4011 {}% no nested par tags
```

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

```
4012 {% yes nest par tags
4013 \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Now have started a paragraph.

```
4014 \global\booltrue{LWR@doingapar}%
```

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

```

4015      \let\par\LWR@closeparagraph%
4016      }% end of yes nest par tags
4017 }% end of handling pars
4018 {}% not handling pars
4019 }

```

\LWR@closeparagraph

```

4020 \newcommand*{\LWR@closeparagraph}
4021 {%
4022 \LWR@traceinfo{\LWR@closeparagraph}%

```

See if paragraph handling is enabled:

```

4023 \ifbool{\LWR@doingapar}%

```

If currently in paragraph mode:

```

4024 {% handling pars

```

See if already started a lateximage or a :

```

4025   \ifboolexpr{
4026       test {\ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}} or
4027       test {\ifnumcomp{\value{\LWR@spandepth}}{>}{0}}
4028   }%

```

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

```

4029   {% no nested par tags
4030       \ifboolexpr{
4031           test {\ifnumcomp{\value{\LWR@spandepth}}{>}{0}} and
4032           test {\ifnumcomp{\value{\LWR@lateximagedepth}}{=}{0}}
4033       }%
4034       {\ifbool{\LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
4035       }%
4036   }% no nested par tags

```

If have not already started a lateximage or a :

```

4037   {% yes nest par tags

```

Print a closing tag and some extra vertical space:

```
4038      \unskip%
4039      \LWR@htmltagc{/\LWR@tagregularparagraph}%
4040      \LWR@orignewline%
```

No longer doing a paragraph:

```
4041      \global\boolfalse{LWR@doingapar}%
4042 % Disable the special \env{minipage} \& \cs{hspace} interaction
4043 % until a new minipage is found:
4044 %   \begin{macrocode}
4045      \global\boolfalse{LWR@minipagethispar}%
4046 %}% end of yes nest par tags
4047}% end of handling pars
```

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

```
4048 {% not handling pars
4049   \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
4050   {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
4051   }%
4052}% not handling pars
```

In most cases, finish with a \LaTeX `\par`, but in the case of paragraphs between lines in a tabular fetch the next token instead:

```
4053 \ifbool{expr}%
4054   not bool {LWR@doingapar} and
4055   test {\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}} and
4056   test {
4057     \ifnumcomp{\value{LWR@tabulardepth}}{=}{\value{LWR@tabularpardepth}}
4058   } and
4059   bool {LWR@intabularmetadata} and
4060   not bool {LWR@tableparcell} and
4061   test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
4062 }%
4063 {%
4064   \LWR@getmynexttoken%
4065 }{%
4066   \LWR@origpar%
4067 }%
4068 }

4069 \end{warpHTML}
```

46 Paragraph start/stop handling

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

Section 45 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: 4070 `\begin{warpHTML}`

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

```
4071 \newcommand*{\LWR@startpars}%
4072 {%
4073 \LWR@traceinfo{\LWR@startpars}%
```

Ignore if inside a span:

```
4074 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
4075 {}%
4076 {%
```

See if currently handling HTML paragraphs:

```
4077 \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
4078 {}%
```

If not currently in paragraph mode:

```
4079 {%
```

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

```
4080 \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
4081 \let\par\LWR@closeparagraph
4082
4083 }% an intentionally blank line
```

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

```
4084 \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
4085 \global\boolfalse{LWR@doingapar}%
4086 }% nestspan
4087 \LWR@traceinfo{LWR@startpars: done}%
4088 }
```

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

```
4089 \newcommand*{\LWR@stoppars}%
4090 {%
```

Ignore if inside a span:

```
4091 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
4092 }%
4093 {%
```

See if currently handling HTML paragraphs:

```
4094 \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
4095 {%
```

Print a closing tag:

```
4096 \unskip%
4097 \LWR@htmltagc{/LWR@tagregularparagraph}%
4098 \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
4099 \global\boolfalse{LWR@doingapar}%
```

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

```
4100 \global\boolfalse{LWR@minipagethispar}
4101
4102 }% an intentionally blank line
```

If was not in an HTML paragraph:

```
4103    {}%
```

See if currently allowing HTML paragraphs:

```
4104    \ifbool{LWR@doingstartpars}%
```

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

```
4105    {\ClearPreHook{par}}%
```

Else: do nothing

```
4106    {}%
```

no longer in paragraph mode

```
4107    \global\setbool{LWR@doingstartpars}{false}%
```

no <p> tag to undo:

```
4108    \global\boolfalse{LWR@doingapar}%
4109 }% nestspan
4110 }
```

```
4111 \end{warpHTML}
```

47 Page headers and footers

for HTML & PRINT: 4112 \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.

```
4113 \newcommand{\LWR@firstpagetop}{} % for the home page alone
4114 \newcommand{\LWR@pagetop}{} % for all other pages
4115 \newcommand{\LWR@pagebottom}{}%
```

\HTMLFirstPageTop {<text and logos>}

```
4116 \newcommand{\HTMLFirstPageTop}[1]{%
4117     \renewcommand{\LWR@firstpagetop}{#1}%
4118 }
```

`\HTMLPageTop` $\{\langle text\ and\ logos\rangle\}$

```
4119 \newcommand{\HTMLPageTop}[1]{%
4120     \renewcommand{\LWR@pagetop}{#1}%
4121 }
```

`\HTMLPageBottom` $\{\langle text\ and\ logos\rangle\}$

```
4122 \newcommand{\HTMLPageBottom}[1]{%
4123     \renewcommand{\LWR@pagebottom}{#1}%
4124 }

4125 \end{warpall}
```

48 CSS

for HTML output: 4126 `\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.

```
4127 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

```
4128 \newcommand*{\CSSFilename}[1]{%
4129 \renewcommand*{\LWR@currentcss}{#1}%
4130 \@onelevel@sanitize\LWR@currentcss%
4131 }
4132
4133 \end{warpHTML}
```

for PRINT output: 4134 `\begin{warpprint}`
 4135 `\newcommand*{\CSSFilename}[1]{}`
 4136 `\end{warpprint}`

49 Title, HTML meta author, HTML meta description

for HTML output: 4137 `\begin{warpHTML}`

`\title` $\{\langle title \rangle\}$ Modified to remember `\thetitle`, which is used to set the HTML page titles.

```
4138 \let\LWR@origtitle\title
4139
4140 \renewcommand*{\title}[1]{%
4141     \LWR@origtitle{#1}%
4142     \begingroup%
4143         \renewcommand{\thanks}[1]{}%
4144         \protected@xdef\thetitle{#1}%
4145     \endgroup%
4146 }
```

```
4147 \end{warpHTML}
```

for HTML & PRINT: 4148 `\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.

```
4149 \providecommand{\thetitle}{}
4150
4151 \newcommand{\theHTMLTitle}{\thetitle}
4152
4153 \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.

```
4154 \providecommand{\theauthor}{}
4155
4156 \newcommand{\theHTMLAuthor}{\theauthor}
4157
4158 \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` $\{\langle New HTML meta description. \rangle\}$ Assigns the HTML file's description meta tag.

```
4159 \newcommand{\LWR@currentHTMLDescription}{}
4160
```

```
4161 \newcommand{\HTMLDescription}[1]{%
4162 \renewcommand{\LWR@currentHTMLDescription}{#1}
4163 }
4164
4165 \end{warpall}
```

50 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.

50.1 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 50.4 for the implementation.

50.2 Minipage footnotes

See section 50.5 for how minipage footnotes are gathered. See section 79.3 for how minipage footnotes are placed into the document.

50.3 Titlepage thanks

See section 57.7 for titlepage footnotes.

50.4 Regular page footnote implementation

for HTML & PRINT: 4166 `\begin{warpall}`

Ctr 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.

```
4167 \newcounter{FootnoteDepth}
4168 \setcounter{FootnoteDepth}{5}
```

```
4169 \end{warpall}
```

for HTML output: 4170 `\begin{warpHTML}`

Patch \TeX footnotes to use a new `\box` instead of an insert for lwarp footnotes. This avoids having the original `\footins` appear at the bottom of a `lateximage`, which is on its own new page.

```
4171 \newbox\LWR@footnotes
```

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

```
\@makefntext {\text}
```

```
4172 \long\def\@makefntext#1{\textsuperscript{\@thefnmark}~#1}
```

```
\@makefnmark
```

```
4173 \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}
```

```
4174 \long\def\LWR@footnotetext#1{%
4175 \LWR@traceinfo{LWR@footnotetext}%
4176 \global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

```
4177 \unvbox\LWR@footnotes%
```

Remember the footnote number for \ref:

```
4178 \protected@edef\@currentlabel{%
4179 \csname p@footnote\endcsname\@thefnmark%
4180 }% @currentlabel
```

Open a group:

```
4181 \color@begingroup%
```

Use HTML superscripts even inside a lateximage:

```
4182 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

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

```
4183 \ifthenelse{%
4184 \boolean{LWR@doingstartpars} \AND%
4185 \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
4186 }%
4187 {}%
4188 {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
4189 \@makefntext{#1}%
```

Closing paragraph tag:

```
4190 \ifthenelse{%
4191 \boolean{LWR@doingstartpars} \AND%
4192 \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
4193 }%
4194 {\par}%
4195 {%
4196 \LWR@htmltagc{/LWR@tagregularparagraph}%
4197 \LWR@orignewline%
4198 }%
```

Close the group:

```
4199 \color@endgroup%
4200 }% vbox
4201 }%
```

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

```
4202 \let\@footnotetext\LWR@footnotetext
```

50.5 Minipage footnote implementation

Patch \LaTeX minipage footnotes to use a new `\box` instead of an insert for lwarp minipage footnotes. This avoids having the original `\@mpfootins` appear at the bottom of a lateximage, which is on its own new page.

```
4203 \newbox\LWR@mpfootnotes
```

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

```
4204 \long\def\@mpfootnotetext#1{%
4205 \LWR@traceinfo{\@mpfootnotetext}%
4206 \global\setbox\LWR@mpfootnotes\vbox{%
4207   \unvbox\LWR@mpfootnotes
4208   \reset@font\footnotesize
4209   \hsize\columnwidth
4210   \@parboxrestore
4211   \protected@edef\@currentlabel
4212   {\csname p@mpfootnote\endcsname\@thefnmark}%
4213   \color@begingroup
```

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

```
4214   \ifthenelse{%
4215     \boolean{LWR@doingstartpars} \AND%
4216     \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
4217   }%
4218   {}%
4219   {\LWR@htmltagc{\LWR@tagregularparagraph}}%

4220   \@makefntext{%
4221     \ignorespaces#1%
4222   }%
```

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

```
4223   \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
4224   {}%
4225   {%
4226     \LWR@htmltagc{/\LWR@tagregularparagraph}%
4227     \LWR@orignewline%
```

```

4228     }%
4229     \color@endgroup%
4230 }% vbox
4231 \LWR@traceinfo{@mpfootnotetext: done}%
4232 }

```

`\thempfootnote` Redefined to remove the `\itshape`, which caused an obscure compiling error in some situations.

```

4233 \AtBeginDocument{
4234 \def\thempfootnote{@alph@c@mpfootnote}
4235 }

```

50.6 Printing pending footnotes

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

```

4236 \newcommand*{\LWR@printpendingfootnotes}{%
4237 \ifvoid\LWR@footnotes\else
4238     \LWR@forcenewpage
4239     \begin{BlockClass}{footnotes}
4240     \LWR@origmedskip
4241     \unvbox\LWR@footnotes
4242     \setbox\LWR@footnotes=\vbox{}
4243     \end{BlockClass}
4244 \fi
4245 }

```

`\LWR@maybeprintpendingfootnotes` `{<depth>}` Used to print footnotes before sections only if formatting for an EPUB or word processor:

```

4246 \newcommand*{\LWR@maybeprintpendingfootnotes}[1]{%
4247 \ifboolexpr{
4248     not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
4249     bool{FormatEPUB} or
4250     bool{FormatWP}
4251 }%
4252 {\LWR@printpendingfootnotes}%
4253 {}%
4254 }

```

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

```

4255 \newcommand*{\LWR@printpendingmpfootnotes}{%

```

```

4256 \ifvoid\LWR@mpfootnotes\else
4257   \LWR@forcenewpage
4258   \begin{BlockClass}{footnotes}
4259   \LWR@origvspace*\baselineskip}
4260   \unvbox\LWR@mpfootnotes
4261   \setbox\LWR@mpfootnotes=\vbox{}
4262   \end{BlockClass}
4263 \fi
4264 }

4265 \end{warpHTML}

```

51 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: 4266 `\begin{warpHTML}`

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

```

4267 \renewcommand{\marginpar}[2] [] {%
4268 \ifbool{FormatWP}%
4269 {%
4270 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{\marginblock}
4271 #2
4272 \end{LWR@BlockClassWP}
4273 }%
4274 {%
4275   \LWR@htmlspanclass{marginpar}{#2}%
4276 }%
4277 }

```

`\marginparBlock` [*<left>*] [*<right>*]

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

HTML version.

```

4278 \newcommand{\marginparBlock}[2] [] {%
4279 \ifbool{FormatWP}%
4280 {%
4281 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
4282 #2
4283 \end{LWR@BlockClassWP}
4284 }%
4285 {%
4286 \begin{BlockClass}[width:2in; float:right; margin:10pt]{marginparblock}
4287 #2
4288 \end{BlockClass}
4289 }%
4290 }

```

`\reversemarginpar`

```
4291 \renewcommand*{\reversemarginpar}{}

```

`\normalmarginpar`

```
4292 \renewcommand*{\normalmarginpar}{}

```

```
4293 \end{warpHTML}

```

for PRINT output: 4294 \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.

```
4295 \LetLtxMacro\marginparBlock\marginpar

```

```
4296 \end{warpprint}

```

52 Splitting HTML files

- Files are split according to FileDepth and CombineHigherDepths.
- Filenames are sanitized by \LWR@filenamenoblanks.

- `\LWR@newhtmlfile` finishes an HTML page, adds a comment to tell where and how to split the file, then starts a new HTML page.

for HTML & PRINT: 4297 `\begin{warpall}`

Ctr **FileDepth** `{\section depth}` determines how deeply to break into new HTML files, similar to `tocdepth`. The default of -5 produces one large HTML file.

```
4298 \newcounter{FileDepth}
4299 \setcounter{FileDepth}{-5}
```

Bool **CombineHigherDepths** Combine higher-level sections together into one file?

```
4300 \newbool{CombineHigherDepths}
4301 \booltrue{CombineHigherDepths}
```

```
4302 \end{warpall}
```

for HTML output: 4303 `\begin{warpHTML}`

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

```
4304 \newcommand*{\LWR@thisfilename}{}
```

\LWR@thisnewfilename The filename being sanitized.

```
4305 \newcommand*{\LWR@thisnewfilename}{}
```

\LWR@filenameno blanks `{\filename}`

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



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

```
4306 \newcommand*{\LWR@filenameno blanks}[1]{%
4307 \begingroup
```

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

```
4308 \LWR@nullfonts%
4309 \renewcommand*{\LWR@htmltagc}[1]{%}
```

Replaces common symbols and short words with hyphens:

```
4310 \LWR@traceinfo{\LWR@filenamenoblanks \#1: !\#1!}%
4311 \edef\LWR@thisnewfilename{\#1}%
4312 \LWR@traceinfo{\LWR@filenamenoblanks edef: !\LWR@thisnewfilename!}%
4313 \fullexpandarg%
```

Convert spaces into hyphens:

```
4314 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
4315 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4316 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
4317 \StrSubstitute{\LWR@thisnewfilename}%
4318     {\LWR@origampersand}{-}[\LWR@thisnewfilename]
4319 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
4320 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4321 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
4322 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
4323 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
4324 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
4325 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
4326 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
4327 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
4328 \StrSubstitute{\LWR@thisnewfilename}%
4329     {\textless}{-}[\LWR@thisnewfilename]
4330 \StrSubstitute{\LWR@thisnewfilename}%
4331     {\textgreater}{-}[\LWR@thisnewfilename]
4332 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]

4333 \StrSubstitute{\LWR@thisnewfilename}{\_}{-}[\LWR@thisnewfilename]

4334 \StrSubstitute{\LWR@thisnewfilename}{\ }{-}[\LWR@thisnewfilename]
4335 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4336 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4337 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4338 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
4339 \StrSubstitute{\LWR@thisnewfilename}%
4340     {\textbackslash}{-}[\LWR@thisnewfilename]
4341 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4342 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4343 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
4344 %     "~{}" for babel
4345 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4346 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
4347 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
```

Convert short words:

```

4348 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
4349 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
4350 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
4351 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
4352 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
4353 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
4354 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
4355 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
4356 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
4357 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
4358 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
4359 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
4360 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
4361 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
4362 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
4363 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
4364 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
4365 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

4366 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
4367 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
4368 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
4369 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
4370 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4371 %      emdash
4372 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
4373 %      endash
4374 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
4375 \endgroup
4376 }

```

LWR@previousautopagelabel **Ctr** Remembers which autopage label was most recently generated. Used to avoid duplicates.

```

4377 \newcounter{\LWR@previousautopagelabel}
4378 \setcounter{\LWR@previousautopagelabel}{-1}

```

\LWR@newautopagelabel **{<pagenumber counter>}**

```

4379 \newcommand*{\LWR@newautopagelabel}[1]{%
4380 \ifnumequal{\value{\LWR@previousautopagelabel}}{\value{page}}{%
4381 {}}% no action if this autopage label has already been defined
4382 {%
4383     \label{autopage-\arabic{#1}}%

```

```

4384 \setcounter{LWR@previousautopagelabel}{\value{page}}
4385 }%
4386 }

```

`\LWR@newhtmlfile` `{\section name}`

Finishes the current HTML page with footnotes, footer, navigation, then starts a new HTML page with an HTML comment telling where to split the page and what the new filename and CSS are, then adds navigation, side TOC, header, and starts the text body.

```

4387 \newcommand*{\LWR@newhtmlfile}[1]{
4388 \LWR@traceinfo{LWR@newhtmlfile: !#1!}

```

At the bottom of the ending file:

```

4389 \LWR@htmlelementclassend{section}{textbody}
4390
4391 \LWR@printpendingfootnotes
4392

```

No footer between files if EPUB:

```

4393 \ifbool{FormatEpub}{
4394 {}
4395 {
4396 \LWR@htmlelement{footer}
4397
4398 \LWR@pagebottom
4399
4400 \LWR@htmlelementend{footer}
4401 }

```

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

```

4402 \ifthenelse{\boolean{FormatEpub}\OR\boolean{FormatWP}}{
4403 {}
4404 {\ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}}

```

End of this HTML file:

```

4405 \LWR@stoppars
4406 \LWR@htmltag{/body}\LWR@orignewline
4407 \LWR@htmltag{/html}\LWR@orignewline
4408 \LWR@orignewpage
4409
4410 \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.

```
4411 \ifbool{FileSectionNames}%
4412 {\LWR@filenamewithoutblanks{#1}}
4413 {\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.

```
4414 \LWR@traceinfo{\LWR@newhtmlfile: about to print start file}%
4415 \LWR@htmlblockcomment{%
4416 |Start file|
4417 \LWR@htmlsectionfilename{\LWR@thisfilename}|
4418 }
```

At the top of the starting file:

```
4419 \LWR@stoppars
4420
4421 \LWR@filestart{ -- #1}% there is an EMdash in front of the #1
4422
```

Track the page numbers:

```
4423 \setcounter{LWR@latestautopage}{\value{page}}%
4424 \LWR@newautopagelabel{LWR@latestautopage}%
```

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

```
4425 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4426 {}
4427 {\LWR@topnavigation}
4428
```

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

```
4429 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4430 {}
4431 {
4432   \LWR@htmlblockcomment{header}
4433
4434   \LWR@pagetop
4435
4436   \LWR@htmlblockcomment{header}
```

```
4437 }
4438
```

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

```
4439 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4440 {}
4441 {\ifcvoid{thetitle}{}\LWR@printthetitle}}
4442
```

No sidetoc if formatting for an EPUB or word processor:

```
4443 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4444 {}
4445 {\LWR@sidetoc}
4446
```

Start of the <tbody>:

```
4447 \LWR@htmlclass{section}{tbody}
4448
```

Keep paragraph tags disabled for now:

```
4449 \LWR@stoppars
4450
4451 \LWR@traceinfo{LWR@newhtmlfile: done}
4452 }
4453 \end{warpHTML}
```

53 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 $\text{pdf\texttt{latex}}$ to generate characters and $\text{pdf\texttt{totext}}$ to read them. If extra symbols appear in the text, it may be that $\text{pdf\texttt{latex}}$ is actually producing a symbol over or under a character, resulting in $\text{pdf\texttt{totext}}$ picking up the accent symbol separately.



$\text{Xe\texttt{LaTeX}}$ and $\text{Lua\texttt{LaTeX}}$ directly support accented section and file names.

for HTML output: 4454 \begin{warpHTML}

53.1 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.

```
4455 \newbool{LWR@forcinghtmlpage}
4456 \boolfalse{LWR@forcinghtmlpage}
4457
4458 \newcommand*{\ForceHTMLPage}{%
4459 \global\booltrue{LWR@forcinghtmlpage}%
4460 }
```

\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.


```
4461 \newbool{LWR@forcinghtmltoc}
4462 \boolfalse{LWR@forcinghtmltoc}
4463
4464 \newcommand*{\ForceHTMLTOC}{%
4465 \global\booltrue{LWR@forcinghtmltoc}%
4466 }
```

```
4467 \end{warpHTML}
```

for PRINT output: 4468 \begin{warpprint}
4469 \newcommand*{\ForceHTMLPage}{}
4470 \newcommand*{\ForceHTMLTOC}{}
4471 \end{warpprint}

for HTML output: 4472 \begin{warpHTML}

53.2 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.

```

4473 \newbool{LWR@mainmatter}
4474 \DeclareDocumentCommand{\mainmatter}{-}{-}{%
4475 \booltrue{LWR@mainmatter}%
4476 }

```

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

```

4477 \DeclareDocumentCommand{\frontmatter}{-}{-}{%
4478 \boolfalse{LWR@mainmatter}%
4479 }

```

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

```

4480 \DeclareDocumentCommand{\backmatter}{-}{-}{%
4481 \boolfalse{LWR@mainmatter}
4482 }

```

53.3 Sectioning support macros

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

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

```

4483 \newcommand*{\LWR@sectionnumber}[1]{%
4484 \InlineClass{sectionnumber}{#1}%
4485 }

```

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

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

Create an autosection tag.

```

4486 \newcommand*{\LWR@createautosec}[1]{%
4487 \LWR@htmltag{#1 id="\LWR@origmbox{autosec-\arabic{page}}"}%
4488 }

```

`\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.

```

4489 \NewDocumentCommand{\LWR@pushoneclose}{m m}{%
4490 \LWR@traceinfo{\LWR@pushoneclose #1}%
4491 \pushclose{#2}{#1}%
4492 }

```

`\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.

```

4493 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%

```

Close any stacked sections up to this new one.

```

4494 \LWR@closeprevious{#1}%

```

Push a new section depth:

```

4495 \LWR@pushoneclose{#1}{#2}%
4496 }

```

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.

```

4497 \newcounter{LWR@prevFileDepth}
4498 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}

```

`\@secCNTformat` $\{\langle sectiontype \rangle\}$

```

4499 \def\@secCNTformat#1{\csname the#1\endcsname\protect\quad}

```

`\simplechapterdelim` Used by `tocbibind` and `anonchap`.

```

4500 \newcommand*\simplechapterdelim{}

```

`\@chapCNTformat` $\{\langle sectiontype \rangle\}$

`\let` to `\@secCNTformat` by default, but may be redefined by `\simplechapter` and `\restorechapter` from `tocbibind` or `anonchap`.

```

4501 \let\@chapCNTformat\@secCNTformat

```

Ctr LWR@currentautosec Records the page number when the section was created. If a math expression is included in the section name, and SVG math is used, the corresponding `lateximage` will cause the page number to change by the time the following autosec label is created.

```
4502 \newcounter{LWR@currentautosec}
```

```
\LWR@section * [{<TOC name>}] {<name>} {<sectiontype>}
```

The common actions for the high-level sectioning commands.

```
4503 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
4504 \LWR@traceinfo{LWR@section |#2| |#3|}%
4505 \LWR@traceinfo{LWR@section: not an empty section}%
4506 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
4507 \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:

```
4508 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4509 \IfBooleanT{#1}{\LWR@traceinfo{LWR@section: starred}}}%
4510 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}}%
4511 \ifthenelse{%
4512   \(%
4513     \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
4514     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
4515     \(\boolean{LWR@forcinghtmlpage}\)\)%
4516   \)%
4517   \AND%
4518   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}}%
4519   \AND%
4520   \(%
4521     \NOT\boolean{CombineHigherDepths}\OR%
4522     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}}%
4523   \)%
4524   \AND%

4525   \(% phantomsection
4526     \NOT\isempty{#3}%
4527     \OR%
4528     \(\NOT\equal{#1}{\BooleanTrue}\)\)%
```

```
4529    \)%
4530 }%
```

If so: start a new HTML file:

```
4531 {% new file
4532    \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
4533    \IfNoValueTF{#2}%
```

If no optional entry

```
4534    {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
4535    {\LWR@newhtmlfile{#2}}%
4536 }% new file
```

Else: No new HTML file:

```
4537 {% not new file
```

Generate a new \TeX page so that toc and index page number points to the section:

```
4538    \LWR@traceinfo{LWR@section: not a new HTML file, about to LWR@orignewpage}%
4539    \LWR@orignewpage%
4540
4541 }% not new file
```

Remember this section's name for \nameref:

```
4542 \IfValueT{#3}{%
4543 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4544 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4545 }%
```

Print an opening comment with the level and the name; ex: “section” “Introduction”

```
4546
4547 \ifbool{HTMLDebugComments}{%
4548    \begingroup%
4549    \LWR@nullfonts%
4550    \LWR@htmlcomment{Opening #4 ‘‘#3’’}%
4551    \endgroup%
```

```
4552 }{}%
4553
```

For inline sections paragraph and subparagraph, start a new paragraph now:

```
4554 \ifthenelse{%
4555     \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4556 }%
4557 {\LWR@startpars}%
4558 {}%
```

Create the opening tag with an autosec:

```
4559 \LWR@traceinfo{LWR@section: about to LWR@createautosec}%
4560 \LWR@createautosec{\csuse{LWR@tag#4}}%
```

```
4561 \setcounter{LWR@currentautosec}{\value{page}}
```

Check if starred:

```
4562 \IfBooleanTF{#1}%
4563 {%
4564 \LWR@traceinfo{LWR@section: starred}%
```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```
4565 \ifbool{LWR@forcinghtmltoc}%
4566 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4567 {%
4568 }% starred
```

Not starred, so step counter and add to TOC:

```
4569 {% not starred
```

Only add a numbered TOC entry if section number is not too deep:

```
4570     \ifthenelse{%
4571         \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4572     }%
4573     {% if secnumdepth
```

If in the main matter, step the counter and add the TOC entry. For article class, lwarp assumes that all is mainmatter.

```
4574     \LWR@traceinfo{LWR@section: about to test main matter}%
```

```

4575     \ifbool{LWR@mainmatter}%
4576     {%
4577         \LWR@traceinfo{LWR@section: yes mainmatter}%
4578         \refstepcounter{#4}%

```

Add main matter numbered TOC entry with the TOC name or the regular name:

```

4579         \LWR@traceinfo{LWR@section: about to addcontentsline}%
4580         \addcontentsline{toc}{#4}%
4581         {%
4582             \protect\numberline{\csuse{the#4}}%
4583             {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4584         }%
4585         \LWR@traceinfo{LWR@section: finished addcontentsline}%
4586     }% end of if main matter

```

If not main matter, add unnumbered TOC name or regular name:

```

4587     {% not main matter
4588         \LWR@traceinfo{LWR@section: no main matter}%
4589         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4590     }% end of not main matter
4591 }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

4592     {%
4593         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4594     }%

```

For part, print the section type:

```

4595     \ifbool{LWR@mainmatter}%
4596     {%
4597         \ifthenelse{%
4598             \(<\cnttest{\csuse{LWR@depth#4}}{<=}%
4599             {\value{secnumdepth}}\>) \AND%
4600             \(<\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\>)%
4601         }%
4602         {\csuse{#4name}~{}}%
4603     }%

```

Print the section number:

```

4604         \LWR@traceinfo{LWR@section: about to print section number}%
4605         \ifthenelse{%
4606             \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4607         }%

```

```

4608      {%
4609          \ifstrequal{#4}{chapter}%
4610          {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
4611          {\protect\LWR@sectionnumber{\@secCNTformat{#4}}}%
4612      }%
4613      {}%
4614      \LWR@traceinfo{LWR@section: finished print section number}%
4615      }{}%
4616 }% end of not starred

```

Print the section name:

```

4617 \LWR@traceinfo{LWR@section: about to print the section name}%
4618 #3%

```

Close the heading tag, such as /H2:

```

4619 \LWR@traceinfo{LWR@section: about to close the heading tag}%
4620 \LWR@htmltag{\csuse{LWR@tag#4end}}%

```

Generate a \LaTeX label:

```

4621 \LWR@traceinfo{LWR@section: about to create the LaTeX label}%
4622 \LWR@newautopagelabel{LWR@currentautosec}%

```

Start paragraph handing unless is an inline paragraph or subparagraph:

```

4623 \ifthenelse{%
4624     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
4625 }%
4626 {\LWR@startpars}%
4627 {}%

```

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.

```

4628 \ifthenelse{%
4629     \NOT\equal{#1}{\BooleanTrue}\OR%
4630     \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
4631 }%
4632 {% not starred

```

```

4633 \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4634 }% not starred
4635 {}%

```

Reset to defaults if not a phantomsection:

```

4636 \ifstrempy{#3}%
4637 {}%
4638 {%
4639 \global\boolfalse{LWR@forcinghtmlpage}%
4640 \global\boolfalse{LWR@forcinghtmltoc}%
4641 }%
4642 %
4643 \LWR@traceinfo{LWR@section: done}%
4644 }

```

53.4 \section and friends

`\part` * [*TOC name*] {*name*}

```

4645 \newcommand{\part@preamble}{}% for koma-script
4646
4647 \DeclareDocumentCommand{\part}{s o m}{%
4648 \LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
4649 \LWR@stoppars%
4650
4651 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4652
4653 \LWR@section{#1}{#2}{#3}{part}%
4654
4655 \part@preamble% for koma-script
4656 \renewcommand{\part@preamble}{}%
4657 }

```

`\chapter` * [*TOC name*] [*heading name*] {*name*}

```

4658 \let\@printcites\relax% for quotchap package
4659
4660 \newcommand{\chapter@preamble}{}% for koma-script
4661
4662 \@ifundefined{chapter}
4663 {}
4664 {%
4665 \DeclareDocumentCommand{\chapter}{s o o m}{%
4666 \IfValueTF{#2}{

```

```

4667 \LWR@traceinfo{chapter #2}%
4668 }{
4669 \LWR@traceinfo{chapter #4}%
4670 }
4671 \LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
4672 \LWR@stoppars%
4673
4674 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4675
4676 \LWR@section{#1}{#2}{#4}{chapter}%
4677
4678 \@printcites% for quotchap package
4679
4680 \chapter@preamble% for koma-script
4681 \renewcommand{\chapter@preamble}{}%
4682 }
4683 }

```

`\section` * [*TOC name*] [*heading name*] {*name*}

```

4684 \DeclareDocumentCommand{\section}{s o m}{%
4685 \IfValueTF{#2}{
4686 \LWR@traceinfo{section #2}%
4687 }{
4688 \LWR@traceinfo{section #4}%
4689 }
4690 \LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
4691 \LWR@stoppars%
4692
4693 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4694
4695 \LWR@section{#1}{#2}{#4}{section}%
4696 }

```

`\subsection` * [*TOC name*] {*name*}

```

4697 \DeclareDocumentCommand{\subsection}{s o m}{%
4698 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
4699 \LWR@stoppars%
4700
4701 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4702
4703 \LWR@section{#1}{#2}{#3}{subsection}%
4704 }

```

`\subsubsection` * [*TOC name*] {*name*}

```

4705 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4706 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
4707 \LWR@stoppars%
4708
4709 \LWR@startnewdepth{\LWR@depthsubsubsection}%
4710 {\LWR@printclosesubsubsection}%
4711
4712 \LWR@section{#1}{#2}{#3}{subsubsection}%
4713 }

```

`\paragraph` * [*TOC name*] {*name*}

```

4714 \DeclareDocumentCommand{\paragraph}{s o m}{%
4715 \LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
4716 \LWR@stoppars%
4717
4718 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4719
4720 \LWR@section{#1}{#2}{#3}{paragraph}%
4721 }

```

`\subparagraph` * [*TOC name*] {*name*}

```

4722 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4723 \LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
4724 \LWR@stoppars%
4725
4726 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4727
4728 \LWR@section{#1}{#2}{#3}{subparagraph}%
4729 }

4730 \end{warpHTML}

```

54 Starting a new file

for HTML & PRINT: 4731 \begin{warpall}

`\HTMLLanguage` Default language for the HTML lang tag.

```

4732 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
4733
4734 \newcommand*{\HTMLLanguage}[1]{%
4735 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
4736 }

```

```
4737 \end{warpall}
```

for HTML output: 4738 \begin{warpHTML}

```
\LWR@filestart {\title_suffix}
```

Creates the opening HTML tags.

```
4739 \newcommand*{\LWR@filestart}[1]{
4740 \LWR@traceinfo{\LWR@filestart}
```

Locally temporarily disable direct-formatting commands:

```
4741 \begingroup
4742 \LWR@traceinfo{\LWR@filestart: A}
4743 \LWR@nullfonts
4744 \LWR@traceinfo{\LWR@filestart: B}
```

Create the page's HTML header:

```
4745 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4746 \LWR@traceinfo{\LWR@filestart: C}
```

The language is user-adjustable:

```
4747 \LWR@htmltag{html lang="\LWR@currentHTMLELanguage"}\LWR@orignewline
```

Start of the meta data:

```
4748 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4749 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

Author:

```
4750 \ifthenelse{\equal{\theHTMLAuthor}{}}%
4751 {}%
4752 {\LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline}}%
```

lwarp is the generator:

```
4753 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /\LWR@orignewline
4754 \LWR@orignewline
```

If there is a description, add it now:

```
4755 \ifdefempty{\LWR@currentHTMLDescription}{-}{-%
4756 \LWR@htmltag{%
4757 meta name="description" content="\LWR@currentHTMLDescription" /}%
4758   \LWR@orignewline
4759 }%
```

Mobile-friendly viewport:

```
4760 \LWR@htmltag{meta name="viewport" %
4761 content="width=device-width, initial-scale=1.0" /}%
4762   \LWR@orignewline
```

IE patch:

```
4763 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline
4764 \LWR@htmltag{%
4765 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"%
4766 \LWR@htmltag{/script}\LWR@orignewline
4767 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4768 \ifthenelse{\equal{\theHTMLTitle}{}}{%
4769 {}%
4770 {\LWR@htmltag{title}\theHTMLTitle#1\LWR@htmltag{/title}\LWR@orignewline}%
```

The page's stylesheet:

```
4771 \LWR@htmltag{%
4772 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4773 \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.

```
4774 \ifbool{mathjax}%
4775 {%
4776   \begingroup%
4777   \LWR@restoreoriglists%
4778   \boolfalse{LWR@verbtags}
4779   \verbatiminput{lwarp_mathjax.txt}%
4780   \booltrue{LWR@verbtags}
4781   \endgroup%
4782   \LWR@stoppars
4783 }% end of mathjax
4784 {}%
```

End of the header:

```
4785 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4786 \LWR@htmltag{body}\LWR@orignewline
4787 \endgroup
4788 \LWR@traceinfo{LWR@filestart: done}
4789 }

4790 \end{warpHTML}
```

55 Starting HTML output

for HTML output: 4791 \begin{warpHTML}

\LWR@LwarpStart Executed at the beginning of the entire document.

```
4792 \catcode'\$=\active
4793 \newcommand*{\LWR@LwarpStart}
4794 {%
4795 \LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
4796 \ifbool{FormatWP}{%
4797   \setcounter{FileDepth}{-5}%
4798   \boolfalse{HTMLDebugComments}%
4799 }{}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
4800 \edef\LWR@strresult{\HomeHTMLFilename}
4801 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
4802 \edef\LWR@strresult{\HTMLFilename}
4803 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
4804 \LWR@origonecolumn%
4805 \LWR@origpagestyle{empty}%
```

Reduce chance of line overflow in verbatim environments:

```
4806 \LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
4807 \LWR@origraggedright%
4808 \LetLtxMacro{\}\{\LWR@endoffline}%
```

Spread the lines for pdftotext to read them well:

```
4809 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4810 \setlength{\parindent}{0pt}
4811 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
4812 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space around the caption in the HTML:

```
4813 \setlength{\belowcaptionskip}{0ex}
4814 \setlength{\abovecaptionskip}{0ex}
```

Redefine the plain page style to be empty when used by index pages:

```
4815 \renewcommand{\ps@plain}{}%
```

<p>\centering</p> <p>\raggedleft</p> <p>\raggedright</p>	<p>Not used in the HTML environment:</p> <pre>4816 \renewcommand*{\centering}{} 4817 \renewcommand*{\raggedleft}{} 4818 \renewcommand*{\raggedright}{}%</pre>
--	---

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```
4819 \LetLtxMacro{\LWR@origtabular}\tabular}
4820 \LetLtxMacro{\LWR@origendtabular}\endtabular}
4821 \LetLtxMacro{\tabular}\LWR@tabular}
4822 \LetLtxMacro{\endtabular}\endLWR@tabular}
```

Float captions:

```
4823 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in `amsmath` environments and is also patched by `cleveref`.

Label in HTML

```
4824 \let\LWR@origltx@label\ltx@label
4825 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
4826 \let\TeX\LWR@TeX
4827 \let\LaTeX\LWR@LaTeX
4828 \let\LuaTeX\LWR@LuaTeX
4829 \let\LuaLaTeX\LWR@LuaLaTeX
4830 \let\XeTeX\LWR@XeTeX
4831 \let\XeLaTeX\LWR@XeLaTeX
4832 \let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
4833 \global\boolfalse{LWR@doingapar}
4834 \global\boolfalse{LWR@doingstartpars}
```

Document and page settings:

```
4835 \mainmatter
4836 \LWR@origpagenumbering{arabic}
```

Start a new HTML file and a header:

```
4837 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4838 \LWR@filestart{}
4839 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
4840 \LWR@htmltag{header}\LWR@orignewline
4841 \LWR@startpars
4842 \LWR@firstpagetop
4843 \LWR@stoppars
4844 \LWR@htmltag{/header}\LWR@orignewline
4845 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4846 \LWR@htmltag{section class="textbody"}
```

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`.

```
4847 \LWR@patchlists
```

Ensure that math mode is active to call lwarp's patches:

```
4848 \catcode'\$=\active
```

Required for \nameref to work with svg math:

```
4849 \immediate\write\@mainaux{\catcode'\string$\active}%
4850 \LetLtxMacro\LWR@syntaxhighlightone$% balance for editor syntax highlighting
```

Allow HTML paragraphs to begin:

```
4851 \LWR@startpars
4852 \LWR@traceinfo{LWR@lwarpStart: done}
4853 }
4854 \catcode'\$=3% math shift until lwarp starts

4855 \end{warpHTML}
```

56 Ending HTML output

for HTML output: 4856 \begin{warpHTML}

`\LWR@requesttoc` $\{ \langle \textit{boolean} \rangle \} \{ \langle \textit{suffix} \rangle \}$ Requests that a toc, lof, or lot be generated.

```
4857 \newcommand*{\LWR@requesttoc}[2]{%
4858 \ifbool{#1}
4859 {
4860     \expandafter\newwrite\csuse{tf@#2}
4861     \immediate\openout \csuse{tf@#2} \jobname.#2\relax
4862 }{}
4863 }
```

`\LWR@LwarpEnd` Final stop of all HTML output:

```
4864 \newcommand*{\LWR@LwarpEnd}
4865 {
4866 \LWR@stoppars
4867 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
4868 \LWR@htmllementclassend{section}{textbody}
```

Print any pending footnotes:

```
4869 \LWR@printpendingfootnotes
```

Create the footer:

```
4870 \LWR@htmlelement{footer}
4871
4872 \LWR@pagebottom
4873
4874 \LWR@htmlelementend{footer}
```

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

Presumably has a table-of-contents.

```
4875 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
4876 {}
4877 {
4878   \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{ }
4879 }
```

```
4880 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4881 \LWR@htmltag{/body}\LWR@orignewline
4882 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4883 \LWR@orignewpage
```




For lateximage commands:

```
4884 \immediate\closeout\LWR@lateximagesfile
4885 }
```

```
4886 \end{warpHTML}
```

57 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 57.8.
affiliation	lwarp provides for the \author macro an additional \affiliation macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using titlingpage's \theauthor in the main text.
reusing titlepage information	The titling package maintains the definitions of \thetitle, \theauthor, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the \thanks and \affiliation removed, and for \author the \and is replaced to generate a simple inline list of authors separated by commas. Note: \theauthor does not work well with authblk unless the traditional \LaTeX syntax is used.
 \theauthor, authblk	
custom titlepages	\printtitle, \printauthor, etc., are provided for use inside a custom titlepage or titlingpage environment, and these retain the \thanks and \affiliation.
\printthanks	\printthanks has been added to force the printing of thanks inside a titlingpage environment when \maketitle is not used.
	Inside a \titlepage or \titlingpage environment, use \thanks instead of \footnote for acknowledgements, etc.

57.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

\author `{\langle author \rangle}` 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` `{\text}`

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

⚠ **`\theauthor`** `\theauthor` does not work well if `authblk` is used. If `\theauthor` is important, it is recommended to use the standard \TeX syntax for `\author`, optionally with `lwarp`'s `\affiliation` macro as well.

⚠ **`affiliations`** After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

`\affiliation` A solution, provide here, is to define a macro `\affiliation` which, during `\maketitle`, starts a new row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to discard its argument, thus printing only the author names when `\author` is later used inline.

57.2 `\if@titlepage`

for HTML & PRINT: 4887 `\begin{warpall}`

`\if@titlepage` Some classes do not provide `\if@titlepage`. In this case, provide it and force it false.

```
4888 \ifcsvoid{@titlepagefalse}{
```

```

4889 \newif\if@titlepage
4890 \@titlepagefalse
4891 \{}

4892 \end{warpall}

```

57.3 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:

```

4893 \begin{warpall}
4894 \providerobustcmd{\affiliation}[1]{}
4895 \end{warpall}

```

for PRINT output:

```

4896 \begin{warpprint}

```

```

4897 \AtBeginEnvironment{titlepage}{
4898 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
4899 }
4900
4901 \AtBeginDocument{
4902 \@ifpackageloaded{titling}{
4903 \AtBeginEnvironment{titlingpage}{
4904 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
4905 }
4906 }{}% titling loaded
4907 }% AtBeginDocument

4908 \end{warpprint}

```

for HTML output:

```

4909 \begin{warpHTML}

```

Env `titlepage` Sets up a `<div>` of class `titlepage`. Provided even for memoir class, since it is used by `\maketitle`.

```

4910 \DeclareDocumentEnvironment{titlepage}{}
4911 {
4912 \renewrobustcmd{\affiliation}[1]{\ \InlineClass{affiliation}{##1}}
4913 \LWR@printpendingfootnotes
4914 \LWR@forcenewpage
4915 \BlockClass{titlepage}

```

```

4916 }
4917 {
4918 \endBlockClass
4919 \LWR@printpendingfootnotes
4920 }

4921 \end{warpHTML}

```

57.4 Printing the thanks

for HTML & PRINT: 4922 \begin{warpall}

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a titlingpage environment when `\maketitle` was not used.

```

4923 \newcommand*{\printthanks}{\@thanks}

4924 \end{warpall}

```

57.5 Printing the title, etc. in HTML

The following are for printing the title, etc. in a titlepage or a titlingpage in HTML:

for HTML output: 4925 \begin{warpHTML}

`\printtitle`

```

4926 \newcommand*{\printtitle}
4927 {
4928 \LWR@stoppars
4929 \LWR@htmltag{\LWR@tagtitle}%
4930 \@title%
4931 \LWR@htmltag{\LWR@tagtitleend}
4932 \LWR@startpars
4933 }

```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```

4934 \newcommand*{\LWR@printthetitle}
4935 {

```

```

4936 \LWR@stoppars
4937 \LWR@htmltag{\LWR@tagtitle}%
4938 \thetitle%
4939 \LWR@htmltag{\LWR@tagtitleend}
4940 \LWR@startpars
4941 }

```

`\printauthor` HTML version.

```

4942 \newcommand*{\printauthor}{

```

The entire author block is contained in a <div> named author:

```

4943 \begin{BlockClass}{author}

```

`\and` finishes one author and starts the next:

```

4944 \renewcommand{\and}{%
4945 \end{BlockClass}
4946 \begin{BlockClass}{oneauthor}
4947 }

```

Individual authors are contained in a <div> named oneauthor:

```

4948 \begin{BlockClass}{oneauthor}
4949 \@author
4950 \end{BlockClass}
4951 \end{BlockClass}
4952 }

```

`\printdate`

```

4953 \newcommand*{\printdate}{%
4954 \begin{BlockClass}{titledate}
4955 \@date
4956 \end{BlockClass}
4957 }

```

```

4958 \end{warpHTML}

```

57.6 Printing the title, etc. in print form

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in print form:

for PRINT output: `4959 \begin{warpprint}`

`\printtitle`

```
4960 \newcommand*{\printtitle}{\Huge\@title}
```

`\printauthor` Print mode.

```
4961 \newcommand*{\printauthor}
4962     {\large\begin{tabular}[t]{c}\@author\end{tabular}}
```

`\printdate`

```
4963 \newcommand*{\printdate}{\small\textit{\@date}}
```

```
4964 \end{warpprint}
```

57.7 \maketitle for HTML output

An HTML <div> of class titlepage is used.

`\thanks` are a form of footnotes used in the title page. See section 50 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4965 `\begin{warpHTML}`

```
4966 \@ifclassloaded{memoir}
4967 {
4968 \newcommand{\LWR@setfootnoteseries}{%
4969     \renewcommand\thefootnote{\@arabic\c@footnote}%
4970 }
4971 }{% not memoir
4972 \if@titlepage
4973 \newcommand{\LWR@setfootnoteseries}{%
4974     \renewcommand\thefootnote{\@arabic\c@footnote}%
4975 }
4976 \else
4977 \newcommand{\LWR@setfootnoteseries}{%
4978     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
4979 }
4980 \fi
4981 }% not memoir
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
4982 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
4983 \LWR@setfootnoteseries%
4984 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

```
\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
4985 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4986 \textsuperscript{\@thefnmark}~%

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
4987 ##1%
4988 }%
4989 }
```

`\@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`.

```
4990 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4991 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
4992 **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
4993 \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.

```
4994 \renewcommand*\maketitle{%
```

An HTML titlepage `<div>` is used for all classes.

```
4995 \begin{titlepage}
```

Set up special patches:

```
4996 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4997 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4998 \@thanks
```

Close the HTML titlepage div and cleanup:

```
4999 \end{titlepage}
5000 \setcounter{footnote}{0}%
5001 \global\let\thanks\relax
5002 \global\let\maketitle\relax
5003 \global\let\@maketitle\relax
5004 \global\let\@thanks\@empty
5005 \global\let\@author\@empty
5006 \global\let\@date\@empty
5007 \global\let\@title\@empty
5008 \global\let\title\relax
5009 \global\let\author\relax
5010 \global\let\date\relax
5011 \global\let\and\relax
5012 }
```

\@maketitle HTML mode. Typesets the title, etc.:

```
5013 \DeclareDocumentCommand{\@maketitle}{-}{%
5014   \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
5015   \@title
5016   \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
5017   \begin{BlockClass}{author}}
```

For IEEEtran class:

```
5018   \renewcommand*{\cr}{-}
5019   \renewcommand*{\crr}{-}
5020   \renewcommand*{\noalign}{-}

5021   \renewcommand{\and}{-}
5022   \end{BlockClass}
5023   \begin{BlockClass}{oneauthor}
5024 }
```

```

5025      \begin{BlockClass}{oneauthor}
5026          \@author
5027      \end{BlockClass}
5028  \end{BlockClass}
5029  \begin{BlockClass}{titledate}
5030      \@date
5031  \end{BlockClass}
5032 }

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```
5033 \newcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
5034 \@thanks
```

Set up special patches:

```
5035 \LWR@maketitlesetup
```

Typeset the title, etc:

```
5036 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
5037 \@thanks
```

```
5038 }
```

```
5039 \end{warpHTML}
```

57.8 `\published` and `\subtitle`

`\subtitle` and `\published` To add `\subtitle` and `\published` to the titlepage, load the titling package and use `\AddSubtitlePublished` in the preamble.

The default `lwarp.css` has definitions for the `published` and `subtitle` classes.

After titling is loaded, `\AddSubtitlePublished` is created, which when used then creates a number of additional macros, and also assigns some of the titling hooks.

titling hooks

Do not use `\AddSubtitlePublished` if the user has patched the titling hooks for some other reason. Portions are marked `\warpprintonly` to reduce extra tags in HTML. Similarly, `BlockClass` has no effect in print mode. Thus, the following may be marked `warpall`.

for HTML & PRINT: 5040 \begin{warpall}

\AddSubtitlePublished Adds \published and \subtitle, and related.

```

5041 \AfterPackage{titling}{
5042 \newcommand*{\AddSubtitlePublished}{%
5043
5044 \newcommand{\@published}{}
5045
5046 \newcommand{\published}[1]{\gdef\@published{##1}}
5047
5048 \renewcommand*{\maketitlehooka}{\printpublished}
5049
5050 \newcommand*{\printpublished}{%
5051 \warpprintonly{\begin{center}\unskip}%
5052 \begin{BlockClass}{published}%
5053 \warpprintonly{\large\itshape}%
5054 \@published%
5055 \end{BlockClass}%
5056 \warpprintonly{\end{center}}}%
5057 }
5058
5059 \newcommand{\@subtitle}{}
5060
5061 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}
5062
5063 \renewcommand*{\maketitlehookb}{\printsubtitle}
5064
5065 \newcommand*{\printsubtitle}{%
5066 \warpprintonly{\begin{center}\unskip}%
5067 \begin{BlockClass}{subtitle}%
5068 \warpprintonly{\Large\itshape}%
5069 \@subtitle%
5070 \end{BlockClass}%
5071 \warpprintonly{\end{center}}}%
5072 }
5073
5074 }% \AddSubtitlePublished
5075 }% AfterPackage

5076 \end{warpall}

```

58 Abstract

The following code replaces the \LaTeX default, and will itself be replaced later if the abstract package is loaded.

for HTML output: 5077 `\begin{warpHTML}`

`\abstractname` User-redefinable title for the abstract.

Also over-written by the babel package.

5078 `\providecommand*{\abstractname}{Abstract}`

Some classes allow an optional name, so it is allowed here.

Env `abstract`

```
5079 \DeclareDocumentEnvironment{abstract}{0{\abstractname}}
5080 {
5081 \LWR@forcenewpage
5082 \BlockClass{abstract}
5083 \BlockClassSingle{abstracttitle}{#1}
5084 }
5085 {
5086 \endBlockClass
5087 }

5088 \end{warpHTML}
```

59 Quote and verse

59.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

ex: `\attribution{author name} --- \citetitle{book name}`

for HTML output: 5089 `\begin{warpHTML}`
 5090 `\newcommand{\attribution}[1]{%`
 5091 `\InlineClass{attribution}{--\, #1}}% emdash`
 5092 `\end{warpHTML}`

for PRINT output: 5093 `\begin{warpprint}`
 5094 `\newcommand{\attribution}[1]{\textsc{---\, #1}}`
 5095 `\end{warpprint}`

`\citetitle` for use inside quote, quotation, verse:

for HTML output:

```

5096 \begin{warpHTML}
5097 \newcommand{\citetitle}[1]{%
5098 \InlineClass{citetitle}{--\,#1}}% emdash
5099 \end{warpHTML}

```

for PRINT output:

```

5100 \begin{warpprint}
5101 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
5102 \end{warpprint}

```

59.2 Quotes, quotations

for HTML output:

```

5103 \begin{warpHTML}

```

Env `quote`

```

5104 \renewenvironment*{quote}
5105 {
5106 \LWR@forcenewpage
5107 \LWR@htmlblocktag{blockquote}
5108 }
5109 {\LWR@htmlblocktag{/blockquote}}
5110
5111 \renewenvironment*{quotation}
5112 {
5113 \LWR@forcenewpage
5114 \LWR@htmlblocktag{blockquotation}
5115 }
5116 {\LWR@htmlblocktag{/blockquotation}}
5117 \end{warpHTML}

```

59.3 Verse

`\attrib` The documentation for the verse and memoir packages suggest defining an `\attrib` command, which may already exist in current documents, but it will only work for print output. `lwarp` provides `\attribution`, which works for both print and HTML output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

```


\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```

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

 **spacing** Horizontal spacing relies on pdftotext’s ability to discern the layout (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

for HTML & PRINT: 5118 `\begin{warpall}`

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

Len `\TMLvleftskip` Sets `\vleftskip` inside a verse environment in HTML.

```
5119 \newlength{\HTMLvleftskip}
5120 \setlength{\HTMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a verse environment in HTML.

```
5121 \newlength{\HTMLleftmargini}
5122 \setlength{\HTMLleftmargini}{4.5em}
```

```
5123 \end{warpall}
```

60 Verbatim and tabbing

for HTML & PRINT: 5124 `\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.

```
5125 \newlength{\VerbatimHTMLWidth}
5126 \setlength{\VerbatimHTMLWidth}{4in}
5127 \end{warpall}
```

for HTML output: 5128 `\begin{warpHTML}`

Bool LWR@verbtags Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
5129 \newbool{LWR@verbtags}
5130 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [`<style>`] [`<class>`]

Encloses a verbatim environment with the given css class.

```
5131 \newcommand*{\LWR@atbeginverbatim}[2] []
5132 {%
```

Avoid excessive space between lines:

```
5133 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
5134 \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.

```
5135 \ifbool{LWR@verbtags}{%
5136   \LWR@htmltag{pre class="#2"
5137   \ifthenelse{\equal{#1}{}}{}{style="#1"}}%
5138 }\LWR@orignewline% pre
5139 }{}%
```

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.

```
5140 \begingroup%
5141 \LWR@origttfamily%
```

Since inside a `<pre>`, restore the original list processing:

```
5142 \LWR@restoreoriglists%
```

Turn off babel-french extra space before punctuation:

```
5143 \LWR@FBcancel%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain \TeX \hspace functionality:

```
5144 \LetLtxMacro{\hspace}{\LWR@orighspace}%
5145 }
```

`\LWR@afterendverbatim` Finishes enclosing a verbatim environment.

```
5146 \newcommand*{\LWR@afterendverbatim}{%
5147
5148 \endgroup%
```

At the end of the environment, close the pre:

```
5149 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
5150
5151 }{}}%
```

Resume regular paragraph handling:

```
5152 \LWR@startpars%
5153 }
```

`\verbatiminput` $\{\langle filename \rangle\}$

Patch `\verbatiminput` to add HTML tags:

```
5154 \let\LWRV@origverbatim@input\verbatim@input
5155
5156 \renewcommand{\verbatim@input}[2]{%
5157 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
5158 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%%
5159 \LWRV@origverbatim@input{#1}{#2}%
5160 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
5161 }
```

Env `verbatim`

```
5162 \AfterEndPreamble{
5163 \LWR@traceinfo{Patching verbatim.}
5164 \AtBeginEnvironment{verbatim}{%
5165 \LWR@forcenewpage%
5166 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-2\baselineskip}%
5167 }
5168 \AfterEndEnvironment{verbatim}{%
5169 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
5170 }
```

5171 }

Env `tabbing` The `tabbing` environment works, except that `svg` math and `lateximages` do not yet work inside the environment.

```

5172 \AfterEndPreamble{
5173 \LWR@traceinfo{Patching tabbing.}
5174 \AtBeginEnvironment{tabbing}{%
5175 \LWR@forcenewpage
5176 \LWR@atbeginverbatim{tabbing}\unskip\LWR@origvspace*{-\baselineskip}%
5177 }
5178 \AfterEndEnvironment{tabbing}{%
5179 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
5180 }
5181 }

5182 \end{warpHTML}

```

61 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: 5183 `\begin{warpHTML}`

`\@begintheorem` $\{\langle name \rangle\}$ $\{\langle number \rangle\}$

```

5184 \renewcommand{\@begintheorem}[2]{%
5185 \LWR@forcenewpage
5186 \BlockClass{theoremcontents}
5187 \trivlist
5188 \item[\InlineClass{theoremlabel}{#1\ #2\ }]\itshape
5189 }

```

`\@opargbegintheorem` $\{\langle name \rangle\}$ $\{\langle number \rangle\}$ $\{\langle oparg \rangle\}$

```

5190 \renewcommand{\@opargbegintheorem}[3]{%
5191 \LWR@forcenewpage
5192 \BlockClass{theoremcontents}
5193 \trivlist
5194 \item[\InlineClass{theoremlabel}{#1\ #2\ (#3)\ }]\itshape
5195 }

```

`\@endtheorem`

```

5196 \renewcommand*{\@endtheorem}{%
5197 \endtrivlist
5198 \endBlockClass% theoremcontents
5199 }

```

```

5200 \end{warpHTML}

```

62 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}

```

`\makelabel` While inside a list environment, `lwarp` nullifies a number of \TeX horizontal skip and fill commands, allowing the user to define `\makelabel` for print mode while HTML mode ignores those commands.

- ⚠ **label font** When defining `\makelabel` in a list environment, use `\textbf` etc. instead of `\bfseries`.

62.1 List environment

for HTML output: 5201 `\begin{warptHTML}`

`\LWR@printcloseslist` May be locally redefined by `enumerate` or `description`.

5202 `\newcommand*{\LWR@printcloseslist}{\LWR@printcloseitemize}`

`\LWR@printopenlist` May be locally redefined by `enumerate` or `description`.

5203 `\newcommand*{\LWR@printopenlist}{\ul style="\LWR@origmbox{list-style-type:none}}}`

`\@mklab` Removes PDF spacing.

```
5204 \AtBeginDocument{
5205 \def\@mklab#1{%
5206 % \hfil %
5207 #1}
5208 \let\makelabel\@mklab
5209 }
```

`\@donoparitem` Modified for HTML output by replacing \TeX boxes with plain text. Also removes PDF spacing.

```
5210 \def\@donoparitem{%
5211 \@noperitemfalse
5212 % \global\setbox\@labels\hbox{\hskip -\leftmargin
5213 % \unhbox\@labels
5214 % \hskip \leftmargin}%
5215 % \if@minipage\else
5216 % \@tempskipa\lastskip
5217 % \vskip -\lastskip
5218 % \advance\@tempskipa\@outerparskip
5219 % \advance\@tempskipa -\parskip
5220 % \vskip\@tempskipa
5221 % \fi
5222 }
```

`\@item` Modified for HTML output by replacing \TeX boxes with plain text. Also removes PDF spacing.

```

5223 \def\LWR@HTML@item[#1]{%
5224 \LWR@traceinfo{@item}
5225   \if@noparitem
5226     \@donoparitem
5227   \else
5228 %     \if@inlabel
5229 %       \indent
5230 %     \par
5231 %     \fi
5232   \ifhmode
5233 %     \unskip\unskip
5234 %     \par
5235   \fi
5236   \if@newlist
5237     \if@nobreak
5238       \@nbitem
5239     \else
5240 %       \addpenalty\@beginparpenalty
5241 %       \addvspace\@topsep
5242 %       \addvspace{-\parskip}%
5243     \fi
5244   \else
5245 %     \addpenalty\@itempenalty
5246 %     \addvspace\itemsep
5247   \fi
5248   \global\@inlabeltrue
5249 \fi
5250 % \everypar{%
5251 % \@minipagefalse
5252 % \global\@newlistfalse

5253 %   \if@inlabel
5254 %     \global\@inlabelfalse

5255 %   {\setbox\z@\lastbox
5256 %     \ifvoid\z@
5257 %       \kern-\itemindent
5258 %     \fi}%

5259 %   \box\@labels
5260 %   \penalty\z@
5261 % \fi

5262 %   \if@nobreak
5263 %     \@nobreakfalse
5264 %     \clubpenalty \@M
5265 %   \else
5266 %     \clubpenalty \@clubpenalty
5267 %     \everypar{}%

```

```

5268 %      \fi}%

5269 \if@noitemarg
5270   \@noitemargfalse
5271   \if@nmbrrlist

5272       \refstepcounter\@listctr
5273   \fi
5274 \fi

5275   \makelabel{#1}%
5276 %   \sbox\@tempboxa{\makelabel{#1}%
5277 %   \global\setbox\@labels\hbox{%
5278 %   \unhbox\@labels
5279 %   \hskip \itemindent
5280 %   \hskip -\labelwidth
5281 %   \hskip -\labelsep
5282 %   \ifdim \wd\@tempboxa >\labelwidth
5283 %       \box\@tempboxa

5284 %   \else
5285 %       \hbox to\labelwidth {\unhbox\@tempboxa}%
5286 %   \fi
5287 %   \hskip \labelsep}%
5288 \ignorespaces%
5289 }

```

`\@nbitem`

```

5290 \def\@nbitem{%
5291 %   \@tempskipa\@outerparskip
5292 %   \advance\@tempskipa -\parskip
5293 %   \addvspace\@tempskipa
5294 }

```

`\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.

```

5295 \newcommand*{\LWR@listitem}{%
5296 \LWR@stoppars%
5297 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
5298 \LWR@htmltag{li}%
5299 \LWR@startpars%

```

```
5300 \LWR@origitem%
5301 }
```

`\LWR@nulllistfills` Nullifies various TeX fill commands, in case they are used inside `\makeatlabel`.

```
5302 \newcommand*{\LWR@nulllistfills}{%
5303 \renewcommand*{\hss}{}}%
5304 \renewcommand*{\llap}[1]{##1}%
5305 \renewcommand*{\rlap}[1]{##1}%
5306 \renewcommand*{\hfil}{}}%
5307 \renewcommand*{\hfilneg}{}}%
5308 \renewcommand*{\hfill}{}}%
5309 }
```

Env `list` `{<label>}{<commands>}`

```
5310 \newcommand*{\LWR@liststart}{%
5311 \LWR@traceinfo{\LWR@liststart}%
5312 \LWR@stoppars%
5313 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
5314 \LWR@htmltag{\LWR@printopenlist}\LWR@orignewline%
5315 \LWR@startpars%
5316 \setlength{\topsep}{0pt}%
5317 \setlength{\partopsep}{0pt}%
5318 \setlength{\itemsep}{0pt}%
5319 \setlength{\parsep}{0pt}%
5320 \setlength{\leftmargin}{0pt}%
5321 \setlength{\rightmargin}{0pt}%
5322 \setlength{\listparindent}{0pt}%
5323 \setlength{\itemindent}{0pt}%
5324 \setlength{\labelsep}{1em}%
5325 \LWR@nulllistfills%
5326 }
```

```
5327 \newcommand*{\LWR@listend}{%
5328 \LWR@traceinfo{\LWR@listend}%
5329 \LWR@stoppars%
5330 \LWR@closeprevious{\LWR@depthlist}%
5331 \LWR@startpars%
5332 }
```

62.2 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.

```
5333 \newcommand*{\LWR@itemizeitem}{%
5334 \LWR@stoppars%
5335 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
5336 \LWR@htmltag{li}%
5337 \LWR@startpars%
5338 \LWR@origitem%
5339 }
```

Env `itemize` [*options*]

```
5340 \newcommand*{\LWR@itemizestart}{%
5341 \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
5342 \renewcommand*{\LWR@printopenlist}{\ul style="\LWR@origmbox{list-style-type:none}"}
5343 \let\item\LWR@itemizeitem%
5344 \LWR@nulllistfills%
5345 }
```

62.3 Enumerate

An HTML unordered list is used with customized \TeX -generated labels.

Env `enumerate` [*options*]

```
5346 \newcommand*{\LWR@enumeratestart}{%
5347 \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
5348 \renewcommand*{\LWR@printopenlist}{\ul style="\LWR@origmbox{list-style-type:none}"}
5349 \let\item\LWR@itemizeitem%
5350 \LWR@nulllistfills%
5351 }
```

62.4 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```
5352 \newcommand*{\LWR@descitem}[1][ ]{%
5353 {%
5354 \LWR@stoppars%
```

```

5355 \LWR@setlatestname{#1}%
5356 \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.

```

5357 \LetLtxMacro{\hspace}{\LWR@nohspace}%

```

Process the original `\item` code:

```

5358 \LWR@origitem[]%

```

Restore `\hspace` for use in the item text:

```

5359 \LetLtxMacro{\hspace}{\LWR@hspace}%
5360 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
5361 \LWR@orignewline%
5362 \LWR@htmltag{dd}%
5363 \LWR@startpars%
5364 }

```

Env description [*options*]

```

5365 \newcommand*{\LWR@descriptionstart}{%
5366 \renewcommand*{\LWR@printcloselist}{\LWR@printclosedescription}
5367 \renewcommand*{\LWR@printopenlist}{dl}
5368 \let\item\LWR@descitem%
5369 \LWR@nulllistfills%
5370 }

```

62.5 Patching the lists

`\LWR@patchlists` Patches list environments.

`\LWR@patchlists` remembers `\item` as defined by whatever packages have been loaded, then patches the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native \TeX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```

5371 \newcommand*{\LWR@patchlists}{%
5372   \LetLtxMacro\item\LWR@listitem%
5373   \LetLtxMacro\@item\LWR@HTML@item%
5374   \renewcommand*{\@trivlist}{%
5375     \LWR@traceinfo{@trivlist start}%
5376     \LWR@liststart%

```

```

5377      \LWR@orig@trivlist%
5378      \LWR@traceinfo{@trivlist done}%
5379  }%
5380  \renewcommand*{\trivlist}{%
5381      \LWR@traceinfo{trivlist}%
5382      \LWR@origtrivlist%
5383  }%
5384  \renewcommand*{\endtrivlist}{%
5385      \LWR@traceinfo{endtrivlist start}%
5386      \LWR@origendtrivlist\LWR@listend%
5387      \LWR@traceinfo{endtrivlist done}%
5388  }%
5389  \renewcommand*{\itemize}{%
5390      \LWR@itemizestart\LWR@origitemize%
5391  }%
5392  \renewcommand*{\enumerate}{%
5393      \LWR@enumeratestart\LWR@origenumerate%
5394  }%
5395  \renewcommand*{\description}{%
5396      \LWR@descriptionstart\LWR@origdescription%
5397  }%
5398 }

```

`\LWR@restoreoriglists` Restores the original trivlist environment.

```

5399 \newcommand*{\LWR@restoreoriglists}{%
5400 \LWR@traceinfo{\LWR@restoreoriglists}%
5401 \LetLtxMacro\item\LWR@origitem%
5402 \LetLtxMacro\@item\LWR@orig@item%
5403 \let\@trivlist\LWR@orig@trivlist%
5404 \let\trivlist\LWR@origtrivlist%
5405 \let\endtrivlist\LWR@origendtrivlist%
5406 \LetLtxMacro\itemize\LWR@origitemize%
5407 \LetLtxMacro\enditemize\LWR@endorigitemize%
5408 \LetLtxMacro\enumerate\LWR@origenumerate%
5409 \LetLtxMacro\endenumerate\LWR@endorigenumerate%
5410 \LetLtxMacro\description\LWR@origdescription%
5411 \LetLtxMacro\enddescription\LWR@endorigdescription%
5412 \let\@mklab\LWR@orig@mklab%
5413 \let\makelabel\LWR@origmakelabel%
5414 \let\@donoparitem\LWR@orig@donoparitem%
5415 \let\@nbitem\LWR@orig@nbitem%
5416 }

5417 \end{warpHTML}

```

63 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

63.1 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
```

⚠ floatrow

This includes before and after defining any macro which used \ttabbox from floatrow.

⚠ tabular inside another
environment

- When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing 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 225.2 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 `warpHTML` environment, if a “misplaced `\noalign`” error occurs, change

```
This & That \endhead
```

to

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

and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.

colortbl:

⚠ `row/cell color`

Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

`colortbl` ignores the overhang arguments.

Other:`longtable headings`

- `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{}`.

⚠ `S columns`

- For `S` columns (from the `siunitx` package), while producing print output, anything non-numeric must be placed inside `{ }` braces, including commands such as `\multirow`. While producing HTML output, though, anything placed inside braces is not seen by `lwarp`'s tabular handling

algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

63.2 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

for HTML output: 5418 `\begin{warpHTML}`

`\LWR@mynexttoken`

5419 `\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`

```
5420 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
5421 \afterassignment\LWR@fns lone\let\nexttoken= }
5422 \def\LWR@fns lone{\expandafter\futurelet\LWR@cs\LWR@fns ltwo}
5423 \def\LWR@fns ltwo{%
5424 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fns lthree%
5425 \else\let\next=\nexttoken\fi\next}
5426 \def\LWR@fns lthree{\afterassignment\LWR@fns lone\let\next= }
```

`\LWR@getmynexttoken` Looks ahead and copies the next token into `\LWR@mynexttoken`.

```
5427 \newcommand*{\LWR@getmynexttoken}{%
5428 \LWR@traceinfo{\LWR@getmynexttoken}%
5429 % nothing must follow this next line
5430 \LWR@futurenonSPACElet\LWR@mynexttoken\LWR@tabledatacolumn tag
5431 }
```

63.3 Tabular variables

Bool `LWR@startedrow` True if should print a row tag before this column.

```
5432 \newbool{LWR@startedrow}
5433 \boolfalse{LWR@startedrow}
```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```
5434 \newbool{LWR@tabularcelladded}
5435 \boolfalse{LWR@tabularcelladded}
```

Bool LWR@doinghline True if the next row will have an hline or midrule above it. Also used for \midrule.

```
5436 \newbool{LWR@doinghline}
5437 \boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
5438 \newbool{LWR@doingtbrule}
5439 \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.

```
5440 \newbool{LWR@doingcmidrule}
5441 \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.

```
5442 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty \multirow cell, and thus there is no data tag to close.

```
5443 \newbool{LWR@skippingmrowcell}
```

Bool LWR@skippingmcolrowcell True if are doing an empty \multicolumn cell, and thus there is no data tag to close, and do not print @ and ! columns.

```
5444 \newbool{LWR@skippingmcolrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.

```
5445 \newbool{LWR@skipatbang}
```

Bool LWR@emptyatbang True if finishing a row and should print empty @ or ! column table data cells.

```
5446 \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.

```
5447 \newbool{LWR@intabularmetadata}
5448 \boolfalse{LWR@intabularmetadata}
```

Ctr LWR@tabularDepth Tracks whether & is being used inside a tabular.

```
5449 \newcounter{LWR@tabulardepth}
5450 \setcounter{LWR@tabulardepth}{0}
```

Ctr LWR@tabularpardepth Tracks whether should look ahead at the next token when encountering a \par while processing tabular contents.

When LWR@tabularpardepth is deeper than LWR@tabulardepth then lwarp has started looking at the contents of the tabular, and thus any \pars encountered must be followed by another token lookahead.

```
5451 \newcounter{LWR@tabularpardepth}
5452 \setcounter{LWR@tabularpardepth}{0}
```

```
5453 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
5454 \newcommand*{\LWR@pposition}{}
5455 \newcommand*{\LWR@pleft}{}
5456 \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 lrrccpc, exactly one letter per \LaTeX table column, without @, !, >, <, or the vertical bar.

```
5457 \newcommand*{\LWR@tablecolspec}{}%
```

\LWR@strresult Holds the result of Str functions.

```
5458 \providecommand*{\LWR@strresult}{}
5459 \providecommand*{\LWR@strresulttwo}{}%
```

\LWR@origcolspec Holds the original column specs given to tabular.

```
5460 \newcommand*{\LWR@origcolspec}{}%
```

Ctr 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 \TeX columns in the table.)

```
5461 \newcounter{LWR@tablecolspecwidth}
```

Ctr LWR@tablecolspecindex While parsing the \TeX 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.

```
5462 \newcounter{LWR@tablecolspecindex}
```

Ctr LWR@tablecolindex While parsing the \TeX 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.

```
5463 \newcounter{LWR@tablecolindex}
```

Ctr LWR@tabletotalcols While parsing a table column specification, begins at 0 and increments by 1 per \TeX table column. Eventually holds the final number of \TeX table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in LWR@tabletotalcols.)

```
5464 \newcounter{LWR@tabletotalcols}
```

Ctr LWR@tabletotalcolsnext Holds the next \TeX table column index while parsing, equal to one more than LWR@tabletotalcols.

```
5465 \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 36.

LWR@colbangspec A data array of specifications for ! columns. The leftmost's index is leftedge, the others are counter values. See section 36.

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.

63.4 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

```

5466 \newcommand*{\LWR@insertatbangcols}{%
5467 \ifbool{\LWR@skipatbang}%
5468 {}%
5469 {%
5470     \LWR@printatbang{at}{\arabic{\LWR@tablecolindex}}%
5471     \LWR@printatbang{bang}{\arabic{\LWR@tablecolindex}}%
5472 }%
5473 }
```

\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.

```

5474 \newcommand*{\LWR@closetabledatacell}{%
5475 \global\booltrue{\LWR@intabularmetadata}%
5476 \ifbool{\LWR@exitingtabular}{}%
5477 {% not exiting tabular
5478     \ifboolexpr{bool{\LWR@skippingmrowcell} or bool{\LWR@skippingmcolrowcell}}%
5479     {%
```

If not skipping a \multicolumnrow cell, insert the @ and ! columns after this non-existent column.

```

5480     \ifbool{\LWR@skippingmcolrowcell}%
5481     {}%
5482     {\LWR@insertatbangcols}%
5483 }%
5484 {% not skippingmrowcell
```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```

5485     \unskip%
5486     \ifboolexpr{%
5487         bool{\LWR@tabularmutemods} or
5488         bool{\LWR@skipatbang} or
5489         bool{\LWR@emptyatbang}
5490     }%
5491     {}%
5492     {\LWR@getexparray{\LWR@colafterspec}{\arabic{\LWR@tablecolindex}}}%

```

Close paragraphs:

```
5493      \ifbool{LWR@tableparcell}{\LWR@stoppars}{}%
5494      \global\boolfalse{LWR@tableparcell}%
```

Close the table data cell.

Close any color <div>s.

```
5495      \whileboolexpr{test {\ifnumcomp{\value{LWR@cellcolordepth}}{>}{0}}}{}%
5496      \LWR@htmltag{/div}\LWR@orignewline%
5497      \addtocounter{LWR@cellcolordepth}{-1}%
5498      }%
```

Skip the @ and ! cells if are closing a multicolumn cell.

```
5499      \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
5500      \global\booltrue{LWR@tabularcelladded}%
5501      \LWR@insertatbangcols%
5502      }% not skipping mrowcell
5503 }% not exiting tabular
5504 \global\boolfalse{LWR@skippingmrowcell}%
5505 \global\boolfalse{LWR@skippingmcolrowcell}%
5506 \global\boolfalse{LWR@skipatbang}%
```

Color control. Column is set by >{} for each cell, so it must be cleared here.

```
5507 \renewcommand*{\LWR@cellHTMLcolor}{}
5508 \renewcommand*{\LWR@columnHTMLcolor}{}
5509 \setcounter{LWR@cellcolordepth}{0}
5510 }
```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```
5511 \let\LWR@origampmacro&

5512 \end{warpHTML}
```

63.4.1 Localizing & catcodes

for HTML & PRINT: 5513 \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.

```
5514 \newcommand{\StartDefiningTabulars}{%
5515 \LWR@traceinfo{StartDefiningTabulars}%
5516 \warpHTMLonly{\catcode'\&=\active}%
5517 }
```

`\EndDefiningTabulars` Place after defining something with & in it.

```
5518 \newcommand{\EndDefiningTabulars}{%
5519 \LWR@traceinfo{EndDefiningTabulars}%
5520 \warpHTMLonly{\catcode'\&=4}%
5521 }
```

```
5522 \end{warpall}
```

63.4.2 Handling &

for HTML output: 5523 `\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.

```
5524 \newcommand*{\LWR@tabularampersand}{%
5525 \LWR@traceinfo{\LWR@tabularampersand}%
5526 \ifnumcomp{\value{\LWR@tabulardepth}}{>}{0}%
5527 {%
```

If not skipping a multirow cell, close the current data cell.

```
5528 \unskip%
5529 \LWR@closetabledatacell%
```

Move to the next column.

```
5530 \addtocounter{\LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5531    \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
5532    \LWR@getmynexttoken%
5533 }%
```

If not inside a tabular, performs the original action:

```
5534 {\LWR@origampmacro}%
5535 }
```

& 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.

63.4.3 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.

```
5536 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
5537 \ifboolexpr{%
5538     not bool {LWR@exitingtabular} or%
5539     bool{LWR@doingtbrule} or%
5540     bool{LWR@doingcmidrule} or%
5541     bool{LWR@doinghline} or%
5542     bool{LWR@startedrow}%
5543 }{%
```

To locally temporarily turn off `LWR@exitingtabular` so that table data tags will still be generated:

```
5544 \begingroup%
```

If generating a final row for the \bottomrule borders, turn off the @, !, <, and > column output:

```
5545 \ifbool{LWR@exitingtabular}{%
5546     \booltrue{LWR@tabularmutemods}%
5547 }{}%
```

Reenable the table data tags until finished with the final row:

```
5548 \global\boolfalse{LWR@exitingtabular}%
```

Generate table data tags and ampersands until the right edge:

```
5549 \whilebool{expr}{%
5550     test {
5551         \ifnumcomp{value{LWR@tablecolindex}}{<}{value{LWR@tabletotalcols}}
5552     } or %
5553     (%
5554         bool{LWR@intabularmetadata} and%
5555         not bool{LWR@tabularcelladded} and%
5556         test {
5557             \ifnumcomp{value{LWR@tablecolindex}}{=}{value{LWR@tabletotalcols}}
5558         }%
5559     )%
5560 }%
5561 {%
5562     \LWR@tabledatasinglecolumn%tag%
```

The following is essentially \LWR@tabularampersand with LWR@emptyatbang added to empty the following cells:

```
5563     \LWR@closetabledatacell%
5564     \addtocounter{LWR@tablecolindex}{1}%
5565     \boolfalse{LWR@tabularcelladded}%
5566     \global\booltrue{LWR@emptyatbang}%
```

Starts the next cell:

```
5567     \ifnumcomp{value{LWR@tablecolindex}}{<}{value{LWR@tabletotalcols}}%
5568     {\LWR@getmynexttoken}%
5569     }{}%
5570 }%
```

Reenable the original LWR@exitingtabular to close the entire table:

```
5571 \endgroup%
5572 \global\boolfalse{LWR@emptyatbang}%
5573 }{}% ifbool{expr
5574 }
```

63.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

\LWR@tabularendofline

```
5575 \NewDocumentCommand{\LWR@tabularendofline}{s o}{%
```

Finish the row:

```
5576 \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}{%
5577 {\LWR@tabularfinishrow}%
5578 {\LWR@closetabledatacell}%
5579 \LWR@htmltag{/tr}\LWR@orignewline%
```

xcolor row color support:

```
5580 \@rowcolor%
```

No longer inside a data cell:

```
5581 \global\booltrue{LWR@intabularmetadata}%
```

Not yet started a table row:

```
5582 \global\boolfalse{LWR@startedrow}%
```

Additional setup:

```
5583 \global\boolfalse{LWR@doinghline}%
5584 \global\boolfalse{LWR@doingtbrule}%
5585 \global\boolfalse{LWR@doingcmidrule}%
5586 \LWR@clearmidrules%
5587 \renewcommand*{\LWR@rowHTMLcolor}{}%
```

Start at first column:

```
5588 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
5589 \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide between single column data tag or a special case:

```
5590 \LWR@getmynexttoken%
5591 }
```

63.6 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
5592 \newcommand*{\LWR@colparameter}{}
```

`\LWR@parseatcolumn` Handles @{text} columns.

```
5593 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
5594 \LWR@traceinfo{at column}%
5595 \addtocounter{\LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
5596 \LWR@traceinfo{about to read the next token:}%
5597 \expandarg%
5598 \StrChar{\LWR@origcolspec}{\arabic{\LWR@tablecolspecindex}}[\LWR@colparameter]
5599 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5600 \LWR@traceinfo{have now read the next token}%
5601 \ifnumcomp{\value{\LWR@tabletotalcols}}{=}{0}%
5602 {% left edge of the table:
5603   \LWR@traceinfo{at the left edge}%
5604   \LWR@setexparray{\LWR@colatspec}{\leftedge}{\LWR@colparameter}%
5605   \LWR@traceinfo{at the left edge: %
5606   \LWR@getexparray{\LWR@colatspec}{\leftedge}}%
5607 }%
5608 {% not at the left edge:
5609   \LWR@traceinfo{not at the left edge}%
5610   \LWR@setexparray{\LWR@colatspec}{\arabic{\LWR@tabletotalcols}}{\LWR@colparameter}%
5611   \LWR@traceinfo{at \arabic{\LWR@tabletotalcols}: %
5612   \LWR@getexparray{\LWR@colatspec}{\arabic{\LWR@tabletotalcols}}}%
5613 }%
5614 \let\LWR@colparameter\relax%
5615 \booltrue{\LWR@validtablecol}%
5616 }
```

`\LWR@parsebangcolumn` Handles `!{text}` columns.

```
5617 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
5618 \LWR@traceinfo{bang column}%
5619 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
5620 \LWR@traceinfo{about to read the next token:}%
5621 \expandarg%
5622 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
5623 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
5624 \LWR@traceinfo{have now read the next token}%
5625 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
5626 {% left edge of the table:
5627   \LWR@traceinfo{at the left edge}%
5628   \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
5629 }%
5630 {% not at the left edge:
5631   \LWR@traceinfo{not at the left edge}%
5632   \LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5633   \LWR@traceinfo{bang \arabic{LWR@tabletotalcols}: \LWR@colparameter!}%
5634 }%
5635 \let\LWR@colparameter\relax%
5636 \booltrue{LWR@validtablecol}%
5637 }
```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```
5638 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
5639 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into `\LWR@colparameter`:

```
5640 \expandarg%
5641 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5642 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
5643 \LWR@setexparray{LWR@colbefore-spec}{\arabic{LWR@tabletotalcolsnext}}{\LWR@colparameter}%
5644 \let\LWR@colparameter\relax%
5645 \booltrue{LWR@validtablecol}%
5646 }
```

\LWR@parseaftercolumn Handles <{text} columns.

```
5647 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
5648 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
5649 \expandarg%
5650 \StrChar{\LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
5651 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
5652 \LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
5653 \let\LWR@colparameter\relax%
5654 \booltrue{LWR@validtablecol}%
5655 }
```

\LWR@parsebarcolumn Handles vertical rules.

```
5656 \newcommand*{\LWR@parsebarcolumn}{%
5657 \LWR@traceinfo{bar column}%
```

Remember the bar at this position:

```
5658 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
5659 {% left edge of the table:
5660   \LWR@setexparray{LWR@colbarspec}{leftedge}{tvertbarl}%
5661 }%
5662 {% not at the left edge:
5663   \LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tabletotalcols}}{tvertbarr}%
5664 }%
5665 \booltrue{LWR@validtablecol}%
5666 }
```

63.7 Parsing ‘l’, ‘c’, or ‘r’ columns

`\LWR@parsenormalcolumn` $\{\langle thiscolumn \rangle\}$

Add to the accumulated column specs, advance counters, and pre-clear another column of at, before, and after specs.

```
5667 \newcommand*{\LWR@parsenormalcolumn}[1]{%
5668 \appto\LWR@tablecolspec{#1}%
5669 \addtocounter{\LWR@tabletotalcols}{1}%
5670 \addtocounter{\LWR@tabletotalcolsnext}{1}%
5671 \LWR@traceinfo{normal column \arabic{\LWR@tabletotalcols}: #1}%
5672 \LWR@setexparray{\LWR@colatspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5673 \LWR@setexparray{\LWR@colbangspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5674 \LWR@setexparray{\LWR@colbeforespec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5675 \LWR@setexparray{\LWR@colafterspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5676 \LWR@setexparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalcolsnext}}{}%
5677 \booltrue{\LWR@validtablecol}%
5678 }
```

63.8 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{\langle thiscolumn \rangle\}$ The width will be ignored.

```
5679 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
5680 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
5681 \addtocounter{\LWR@tablecolspecindex}{1}%
5682 }
```

63.9 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\{\langle thiscolumn \rangle\}$ The three parameters will be ignored.

```
5683 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Table 8: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Converts to the given column type.

```
5684 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
5685 \addtocounter{LWR@tablecolspecindex}{3}%
5686 }
```

63.10 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.

```
5687 \newbool{LWR@validtablecol}
```

Bool LWR@opttablecol True if found a table column optional argument.

```
5688 \newbool{LWR@opttablecol}
```

\LWR@parsetablecols $\{\langle colspecs \rangle\}$

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.

Table 9: Tabular HTML column conversions

l, r, c:	Converted to table cells without paragraph tags. Uses css <code>vertical-align:middle</code> so that top or bottom-aligned cells may go above or below this cell.
p:	Converted to table cells with paragraph tags. Ref: Table 8, \LaTeX places the top line of a parbox aligned with the rest of the text line, so css <code>vertical-align:bottom</code> is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the css may be desired, changing <code>td.tdp</code> , <code>td.tdP</code> , <code>td.tdprule</code> , and <code>td.tdPrule</code> to <code>vertical-align: middle</code> . Another possibility is to change L,R,C, and P to <code>vertical-align: top</code> and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
m:	With paragraph tags, css <code>vertical-align:middle</code> .
b:	With paragraph tags, css <code>vertical-align:top</code> so that the bottom of the text is closest to the middle of the text line.
P, M, B:	Horizontally-centered versions.
S:	Converted to 'r'. Ignores optional argument. From the <code>siunitx</code> package.
D:	Converted to 'c'. From the <code>dcolumn</code> package.
@, !, >, <:	One each, in that order.
 :	Vertical rule.
Unknown:	Converted to 'l'.
\newcolumn:	Currently treated as unknown.

```
5689 \newcommand*{\LWR@parsetablecols}[1]{%
5690 \LWR@traceinfo{\LWR@parsetablecols}%
```

Remember the original supplied column spec:

```
5691 \renewcommand*{\LWR@origcolspec}{#1}%
```

Remove spaces:

```
5692 \expandarg%
5693 \StrSubstitute{\LWR@origcolspec}{ }{}[\LWR@origcolspec]%
```

Clear the parsed resulting column spec:

```
5694 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
5695 \setcounter{\LWR@tabletotalcols}{0}%
5696 \setcounter{\LWR@tabletotalcolsnext}{1}%
5697 \LWR@setexparray{\LWR@colatspec}{leftedge}{}%
5698 \LWR@setexparray{\LWR@colatspec}{1}{}%
5699 \LWR@setexparray{\LWR@colatspec}{2}{}%
5700 \LWR@setexparray{\LWR@colatspec}{3}{}%
5701 \LWR@setexparray{\LWR@colbangspec}{leftedge}{}%
5702 \LWR@setexparray{\LWR@colbangspec}{1}{}%
5703 \LWR@setexparray{\LWR@colbangspec}{2}{}%
5704 \LWR@setexparray{\LWR@colbangspec}{3}{}%
5705 \LWR@setexparray{\LWR@colbeforespec}{1}{}%
5706 \LWR@setexparray{\LWR@colbeforespec}{2}{}%
5707 \LWR@setexparray{\LWR@colbeforespec}{3}{}%
5708 \LWR@setexparray{\LWR@colafterspec}{1}{}%
5709 \LWR@setexparray{\LWR@colafterspec}{2}{}%
5710 \LWR@setexparray{\LWR@colafterspec}{3}{}%
5711 \LWR@setexparray{\LWR@colbarspec}{leftedge}{}%
5712 \LWR@setexparray{\LWR@colbarspec}{1}{}%
5713 \LWR@setexparray{\LWR@colbarspec}{2}{}%
5714 \LWR@setexparray{\LWR@colbarspec}{3}{}%
```

Starting at the first column specification:

```
5715 \setcounter{\LWR@tablecolspecindex}{1}%
```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```
5716 \expandarg%
```

```

5717 \StrLen{\LWR@origcolspec}[\LWR@strresult]%
5718 \fullexpandarg%
5719 \LWR@traceinfo{original column spec length: \LWR@strresult}%
5720 \setcounter{\LWR@tablecolspecwidth}{\LWR@strresult}%

```

Haven't seen any optional arguments so far

```

5721 \boolfalse{\LWR@opttablecol}%

```

Scan through the column specifications:

```

5722 \whilebool{expr}{%
5723     not test{%
5724         \ifnumcomp{\value{\LWR@tablecolspecindex}}{>}{\value{\LWR@tablecolspecwidth}}{%
5725             }%
5726 }%
5727 {%

```

Place the next single-character column type into \LWR@strresult:

```

5728 \expandarg%
5729 \StrChar{\LWR@origcolspec}{\arabic{\LWR@tablecolspecindex}}[\LWR@strresult]%
5730 \LWR@traceinfo{position \arabic{\LWR@tablecolspecindex}: \LWR@strresult}%
5731 \fullexpandarg%

```

Not yet found a valid column type:

```

5732 \boolfalse{\LWR@validtablecol}%

```

Skip over any optional arguments, such as siunitx S column:

```

5733 \IfStrEq{\LWR@strresult}{[]}{\booltrue{\LWR@opttablecol}}{%

```

Throw away anything found inside the optional argument:

```

5734 \ifbool{\LWR@opttablecol}%
5735 {}% inside an optional argument
5736 {% not an optional tabular argument

```

Not inside an optional argument, so consider the column type:

```

5737 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{%
5738 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{%
5739 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{%
5740 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{%
5741 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{%
5742 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{%
5743 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{%

```

```

5744 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
5745 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
5746 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
5747 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
5748 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
5749 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
5750 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
5751 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
5752 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

```

From the dcolumn package:

```

5753 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%

```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```

5754 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%

```

Many people define centered versions “P”, “M”, and “B”:

```

\newcolumntype{P}[1]{>\centering\arraybackslashp{#1}}

```

```

5755 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
5756 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
5757 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to an l column:

```

5758 \ifbool{LWR@validtablecol}{}{}%
5759   \LWR@traceinfo{invalid column type: \LWR@strresult}%
5760   \LWR@parsenormalcolumn{l}%
5761 }%
5762 }% not an optional column argument

```

If read the closing bracket, no longer inside the optional argument:

```

5763 \IfStrEq{\LWR@strresult}{]}{\boolfalse{LWR@opttablecol}}{}%

```

Move to the next character:

```

5764 \addtocounter{LWR@tablecolspecindex}{1}%
5765 }% whiledo
5766 \LWR@traceinfo{LWR@parsetablecols: done}%
5767 }%

```

63.11 colortbl and xparse tabular color support

These macros provide a minimal emulation of some colortbl macros which might appear between table cells. If colortbl is loaded, these macros will be replaced with functional versions.

For each of the HTML colors below, the text for the HTML color is set if requested, but the macro is empty if none has been set.

`\rownum` Reserve a counter register.

5768 `\@ifundefined{rownum}{\newcount\rownum}{}`

`\@rowcolors` Emulated in case xcolor is not used.

5769 `\newcommand*\@rowcolors{}`

`\@rowc@lors` Emulated in case xcolor is not used.

5770 `\newcommand*\@rowc@lors{}`

`\LWR@xcolorrowHTMLcolor` Emulated xcolor row color.

5771 `\newcommand*\LWR@xcolorrowHTMLcolor{}`

`\LWR@columnHTMLcolor` HTMLstyle code for the column color.

5772 `\newcommand*\LWR@columnHTMLcolor{}`

`\LWR@rowHTMLcolor` HTMLstyle code for the row color.

5773 `\newcommand*\LWR@rowHTMLcolor{}`

`\LWR@cellHTMLcolor` HTMLstyle code for the cell color.

5774 `\newcommand*\LWR@cellHTMLcolor{}`

`\LWR@ruleHTMLcolor` HTMLstyle code for the cell color.

5775 `\newcommand*\LWR@ruleHTMLcolor{}`

Inside an HTML tabular, each of `\columncolor` etc. is `\let` to the `\LWR@HTML` versions below. When colortbl is loaded, its definitions override the following.

`\columncolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

`\LWR@HTMLcolumncolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

5776 `\NewDocumentCommand{\LWR@HTMLcolumncolor}{0{named} m o o}{}`

`\rowcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

`\LWR@HTMLrowcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*] Used before starting a tabular data cell, thus `\LWR@getmynexttoken`.

5777 `\NewDocumentCommand{\LWR@HTMLrowcolor}{0{named} m o o}{\LWR@getmynexttoken}`

`\cellcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

`\LWR@HTMLcellcolor` [*⟨model⟩*] {*⟨color⟩*} [*⟨left overhang⟩*] [*⟨right overhang⟩*]

5778 `\NewDocumentCommand{\LWR@HTMLcellcolor}{0{named} m o o}{}`

`\arrayrulecolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use outside a tabular.

5779 `\newcommand{\arrayrulecolor}[2][named]{}`

`\LWR@HTMLarrayrulecolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use inside a tabular.

5780 `\newcommand{\LWR@HTMLarrayrulecolor}[2][named]{\LWR@getmynexttoken}`

`\doublerulesepcolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use outside a tabular.

5781 `\newcommand{\doublerulesepcolor}[2][named]{}`

`\LWR@HTMLdoublerulesepcolor` [*⟨model⟩*] {*⟨color⟩*}

The version for use inside a tabular.

5782 `\newcommand{\LWR@HTMLdoublerulesepcolor}[2][named]{\LWR@getmynexttoken}`

63.12 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.

```
5783 \newcommand*{\LWR@maybenewtablerow}
5784 {%
5785 \ifbool{\LWR@startedrow}%
5786 {}% started the row
5787 {% not started the row
```

Remember that now have started the row:

```
5788 \global\booltrue{\LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
5789 \global\booltrue{\LWR@intabularmetadata}%
5790 \ifbool{\LWR@doinghline}%
5791 {%
5792 \ifdefvoid{\LWR@ruleHTMLcolor}{%
5793 \LWR@htmltag{tr class="hline" }%
5794 }{%
5795 \LWR@htmltag{%
5796 tr class="hline" %
5797 style="border-top: 1px solid \#\LWR@ruleHTMLcolor "%
5798 }%
5799 }%
5800 \LWR@orignewline%
5801 }%
5802 {% not doing hline
5803 \ifbool{\LWR@doingtbrule}%
5804 {%
5805 \ifdefvoid{\LWR@ruleHTMLcolor}{%
5806 \LWR@htmltag{tr class="tbrule"}%
5807 }{%
5808 \LWR@htmltag{%
5809 tr class="tbrule" %
5810 style="border-top: 1px solid \#\LWR@ruleHTMLcolor "%
5811 }%
5812 }%
5813 \LWR@orignewline%
5814 }%
5815 {\LWR@htmltag{tr}\LWR@orignewline}%
5816 }% end of not doing hline
5817 }% end of not started the row
5818 }
```

63.13 Printing vertical bar tags

`\LWR@printbartag` $\{\langle index \rangle\}$

Adds to a tabular data cell an HTML class name for a left/right vertical bar.

```

5819 \newcommand*\LWR@printbartag[1]{%
5820 \LWR@traceinfo{\LWR@printbartag !#1!}%
5821 \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}%
5822 {}% muting or empty
5823 {% not muting
5824     \edef\LWR@tempone{\LWR@getexparray{\LWR@colbarspec}{#1}}%
5825     \ifdefempty{\LWR@tempone}{\LWR@tempone}%
5826 }% not muting
5827 \LWR@traceinfo{\LWR@printbartag done}%
5828 }
```

63.14 Printing at or bang tags

`\LWR@printatbang` $\{\langle at-or-bang \rangle\} \{\langle index \rangle\}$

```

5829 \newcommand*\LWR@printatbang[2]{%
```

Fetch the column at or bang spec:

```

5830 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@col#1spec}{#2}}%
5831 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```

5832 \ifdefempty{\LWR@atbangspec}%
5833 {}%
5834 {% not empty
5835     \LWR@htmltag{%
5836         td class="td#1%
5837         \LWR@subaddcmidruletrim{}{}%
5838         \LWR@printbartag{#2}%
5839         "%
5840         \LWR@tdstartstyles%
5841         \LWR@addcmidrulewidth%
5842         \LWR@addtabularrulecolors%
5843         \LWR@tdendstyles%
5844     }%
```

Create an empty cell if muting for the \bottomrule:

```

5845 \ifbool{expr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}}%
5846 {}%
5847 {\LWR@atbangspec}%
5848 %
5849 \LWR@htmltag{/td}\LWR@orignewline%
5850 \global\booltrue{LWR@tabularcelladded}%
5851 }% not empty
5852 }%
```

\LWR@addleftmostbartag

```

5853 \newcommand*{\LWR@addleftmostbartag}{%
5854 \ifnumcomp{\value{LWR@tablecolindex}}{=}{1}{%
5855 \LWR@printbartag{leftedge}%
5856 }{}%
5857 }
```

\LWR@tabularleftedge

```

5858 \newcommand*{\LWR@tabularleftedge}{%
5859 \ifnumcomp{\value{LWR@tablecolindex}}{=}{1}%
5860 {%
5861 \LWR@printatbang{at}{leftedge}%
5862 \LWR@printatbang{bang}{leftedge}%
5863 }% left edge
5864 {}% not left edge
5865 }
```

63.15 Data opening tag

\LWR@thiscolspec Temporary storage.

```

5866 \newcommand*{\LWR@thiscolspec}{}
```

\LWR@tabledatasinglecolumnntag Print a table data opening tag with style for alignment and color.

```

5867 \newcommand*{\LWR@tabledatasinglecolumnntag}%
5868 {%
5869 \LWR@traceinfo{LWR@tabledatasinglecolumnntag}%
5870 \LWR@maybenewtablerow%
```

Don't start a new paragraph tag if have already started one:

```
5871 \ifbool{LWR@intabularmetadata}%
5872 {%
```

If have found the end of tabular command, do not create the next data cell:

```
5873 \ifbool{LWR@exitingtabular}{}%
5874 {% not exiting tabular
```

Print the @ and ! contents before first column:

```
5875 \LWR@tabularleftedge%
```

Fetch the current column's alignment character into \LWR@strresult:

```
5876 \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@strresult]%
```

print the start of a new table data cell:

```
5877 \LWR@traceinfo{LWR@tabledatasinglecolumn tag: about to print td tag}%
5878 \LWR@htmltag{td class="td%
```

append this column's spec:

```
5879 \LWR@strresult%
```

If this column has a cmidrule, add "rule" to the end of the HTML class tag. Also add vertical bar tags.

```
5880 \LWR@addcmidruletrim%
5881 \LWR@addleftmostbartag%
5882 \LWR@printbartag{\arabic{LWR@tablecolindex}}%
5883 "%
```

Add styles for rules, alignment:

```
5884 \LWR@tdstartstyles%
5885 \LWR@addcmidrulewidth%
5886 \StrChar{\LWR@tablecolspec}{\arabic{LWR@tablecolindex}}[\LWR@thiscolspec]%
5887 \LWR@addformatwpalignment{\LWR@thiscolspec}%
```

Add styles for cell and rule colors:

```
5888 \LWR@addtabularrowcolor%
5889 \LWR@addtabularrulecolors%
```

```

5890      \LWR@tdendstyles%
5891      }%
5892      \LWR@traceinfo{LWR@tabledatasinglecolumn tag: done printing td tag}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

5893      \ifboolexpr{%
5894          test{ \ifdefstring{LWR@strresult}{p} } or
5895          test{ \ifdefstring{LWR@strresult}{m} } or
5896          test{ \ifdefstring{LWR@strresult}{b} } or
5897          test{ \ifdefstring{LWR@strresult}{P} } or
5898          test{ \ifdefstring{LWR@strresult}{M} } or
5899          test{ \ifdefstring{LWR@strresult}{B} } or
5900          test{ \ifdefstring{LWR@strresult}{X} }
5901      }%
5902      {% allow pars
5903          \LWR@traceinfo{LWR@tabledatasinglecolumn tag: about to LWR@startpars}%
5904          \global\booltrue{LWR@tableparcell}%
5905          \LWR@startpars%
5906          \LWR@traceinfo{LWR@tabledatasinglecolumn tag: done with LWR@startpars}%
5907      }% allow pars
5908      }% no pars

```

Print the > contents unless muted for the \bottomrule:

```

5909      \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
5910      {%
5911      {%
5912          \LWR@getexparray{LWR@colbefore spec}{\arabic{LWR@tablecolindex}}%
5913      }%
5914      \global\boolfalse{LWR@intabularmetadata}%
5915      }% not exiting tabular
5916 }{}% in tabular metadata
5917 \LWR@traceinfo{LWR@tabledatasinglecolumn tag: done}%
5918 }%

```

63.16 Midrules

LWR@midrules LWR@midrules is a data array (section 36) 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 36) of columns containing 1 if a midrule should be left trimmed for each column.

LWR@trimrrules LWR@trimrrules is a data array (section 36) 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.

```

5919 \newcounter{LWR@midrulecounter}

```

Len \LWR@heavyrulewidth The default width of the rule.

```

5920 \newlength{\LWR@heavyrulewidth}
5921 \setlength{\LWR@heavyrulewidth}{.08em}

```

Len \LWR@lightrulewidth The default width of the rule.

```

5922 \newlength{\LWR@lightrulewidth}
5923 \setlength{\LWR@lightrulewidth}{.05em}

```

Len \LWR@cmidrulewidth The default width of the rule.

```

5924 \newlength{\LWR@cmidrulewidth}
5925 \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.

```

5926 \newlength{\LWR@thiscmidrulewidth}
5927 \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.

```

5928 \newcommand*{\LWR@clearmidrules}
5929 {%
5930 \setcounter{LWR@midrulecounter}{1}%
5931 \whileboolexpr{%
5932     not test{%
5933         \ifnumcomp{\value{LWR@midrulecounter}}{>}{\value{LWR@tablecolspecwidth}}%
5934     }%
5935 }%
5936 {%
5937 \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{Opt}%
5938 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
5939 \LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
5940 \LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
5941 \addtocounter{LWR@midrulecounter}{1}%
5942 }%
5943 }

```

`\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.

```
5944 \newcommand*{\LWR@subcmidrule}[4]{%
5945 \setcounter{LWR@midrulecounter}{#3}%
5946 \whilebool{expr}%
5947     not test {%
5948         \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}%
5949     }%
5950 }%
5951 {%
5952     \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
5953     \addtocounter{LWR@midrulecounter}{1}%
5954 }% whiledo
5955 \IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
5956 \IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
5957 \booltrue{LWR@doingcmidrule}%
5958 }
```

`\LWR@docmidrule` $[\langle width \rangle] \{\langle trim \rangle\} \{\langle leftcolumn-rightcolumn \rangle\}$

Marks `LWR@midrules` array elements to be a non-zero width from left to right columns. Also marks trimming for the L and/or R columns.

```
5959 \NewDocumentCommand{\LWR@docmidrule}{0{\LWR@cmidrulewidth} D{} >\SplitArgument{1}{-}m}%
5960 {\LWR@subcmidrule{#1}{#2}{#3}}
```

Used to compute margins, tabular trims:

```
5961 \newlength{\LWR@templengthone}%
5962 \newlength{\LWR@templengthtwo}%
5963 \newlength{\LWR@templengththree}%
```

Used to add a style to a table data cell:

```
5964 \newboolean{LWR@tdhavecellstyle}
```

`\LWR@tdstartstyles` Begins possibly adding a table data cell style.

```
5965 \newcommand*{\LWR@tdstartstyles}{\global\boolfalse{LWR@tdhavecellstyle}}
```

`\LWR@tdaddstyle` Starts adding a table data cell style.

```
5966 \newcommand*{\LWR@tdaddstyle}{%
5967 \ifbool{\LWR@tdhavecellstyle}%
5968 {; }%
5969 { style="%
5970 \booltrue{\LWR@tdhavecellstyle}%
5971 }
```

`\LWR@tdendstyles` Finishes possibly adding a table data cell style. Prints the closing quote.

```
5972 \newcommand*{\LWR@tdendstyles}{%
5973 \ifbool{\LWR@tdhavecellstyle}{%
5974     "%
5975     \global\boolfalse{\LWR@tdhavecellstyle}%
5976 }{}}%
5977 }
```

`\LWR@subaddcmidruletrim` `{<lefttrim>}{<righttrim>}` Adds a `\cmidrule` with optional trim.

```
5978 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
5979 \setlength{\LWR@templengthone}{%
5980     \LWR@getexparray{\LWR@midrules}{\arabic{\LWR@tablecolindex}}}%
5981 }%
5982 \ifdimcomp{\LWR@templengthone}{>}{0pt}%
5983 {%
```

Print the class without left and right trim letters appended:

```
5984     \LWR@origtilde tdrule#1#2%
```

Remember the width of the rule:

```
5985     \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
5986 }%
5987 {%
5988     \setlength{\LWR@thiscmidrulewidth}{0pt}%
5989 }%
5990 }
```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```
5991 \newcommand*{\LWR@addcmidruletrim}{%
5992 \LWR@subaddcmidruletrim%
5993 {\LWR@getexparray{\LWR@trimlrules}{\arabic{\LWR@tablecolindex}}}%
5994 {\LWR@getexparray{\LWR@trimrrules}{\arabic{\LWR@tablecolindex}}}%
5995 }
```

`\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`.

```
5996 \newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`, or if a color is being used:

```
5997 \ifboolexpr{%
5998     test{\ifdimcomp{#1}={}\{0pt\}} or
5999     (
6000         ( test{\ifdimcomp{#1}={}\{#2\}} and not bool{FormatWP} )
6001         and ( test {\ifdefvoid{\LWR@ruleHTMLcolor}} )
6002     )
6003 }%
6004 {}% default width and color
6005 {}% custom width and/or color
```

Ensure that the width is wide enough to display in the browser:

```
6006     \LWR@forceminwidth{#1}%
```

Begin adding another style:

```
6007     \LWR@tdaddstyle%
```

The style itself:

```
6008     border-top:\LWR@printlength{\LWR@atleastonept} solid %
```

If default gray, the darkness of the color depends on the thickness of the rule:

```
6009     \ifdefvoid{\LWR@ruleHTMLcolor}{%
6010         \ifdimcomp{#1}{<}\{\LWR@lightrulewidth}%
6011         {\#AOA0A0}%
6012         {% lightrule or heavier
6013             \ifdimcomp{#1}{<}\{\LWR@heavyrulewidth}%
6014             {\#808080}%
6015             {black}%
6016         }% lightrule or heavier
6017     }{%
6018         \#\LWR@ruleHTMLcolor%
6019     }
6020 }% custom width and/or color
```

6021 }

`\LWR@addcmidrulewidth` Adds a style for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6022 \newcommand{\LWR@addcmidrulewidth}{%
6023 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
6024 }
```

`\LWR@WPcell` $\{\langle text-align \rangle\} \{\langle vertical-align \rangle\}$

```
6025 \newcommand*{\LWR@WPcell}[2]{%
6026 \LWR@tdaddstyle%
6027 \LWR@origmbox{text-align:#1}; \LWR@origmbox{vertical-align:#2}%
6028 }
```

`\LWR@addformatwpalignment` If `FormatWP`, adds a style for the alignment.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6029 \newcommand*{\LWR@addformatwpalignment}[1]{%
6030 \ifbool{FormatWP}{%
6031 \IfSubStr{#1}{l}{\LWR@WPcell{left}{middle}}{}%
6032 \IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
6033 \IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%
6034 \IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
6035 \IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
6036 \IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
6037 \IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
6038 \IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
6039 \IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
6040 }{}%
6041 }
```

63.17 Cell colors

`\LWR@addtabulararrowcolor` Adds a cell's row color style, if needed.

No color is added for the final row of empty cells which finishes each tabular.

```
6042 \newcommand*{\LWR@addtabulararrowcolor}{%
6043 \ifbool{LWR@tabularmutemods}{}{}%
6044 \ifdefined{\LWR@rowHTMLcolor}{%
```

```

6045     \ifdefvoid{\LWR@xcolorrowHTMLcolor}{}%
6046     {% xcolor row color
6047         \LWR@tdaddstyle%
6048         background:\#\LWR@xcolorrowHTMLcolor%
6049     }%
6050 }%
6051 {% explicit row color
6052     \LWR@tdaddstyle%
6053     background:\#\LWR@rowHTMLcolor%
6054 }%
6055 }%
6056 }

```

`\LWR@addtabularhrulecolor` Adds a cell's horizontal rule color style, if needed.

```
6057 \newcommand*{\LWR@addtabularhrulecolor}{%
```

If either form of horizontal rule is requested:

```

6058 \ifboolexpr {
6059     bool{\LWR@doinghline} or
6060     bool{\LWR@doingtbrule}
6061 }{%

```

And if there is a custom horizontal color:

```

6062     \ifdefvoid{\LWR@ruleHTMLcolor}{}%
6063     {%
6064         \LWR@tdaddstyle%
6065         border-top: 1px solid \#\LWR@ruleHTMLcolor%
6066     }{}%
6067 }{}%
6068 }

```

`\LWR@addtabularrulecolors` Adds a cell's rule color styles, if needed.

No color is added for the final row of empty cells which finishes each tabular.

```
6069 \newcommand*{\LWR@addtabularrulecolors}{%
```

Custom horizontal rule color:

```
6070 \LWR@addtabularhrulecolor%
```

No vertical rules if finishing the tabular with a row of empty cells:

```
6071 \ifbool{\LWR@tabularmutemods}{}%

```

If at the leftmost cell, possibly add a leftmost vertical rule:

```
6072 \ifnumequal{\value{LWR@tablecolindex}}{1}{%
```

Fetch the left edge's vertical bar specification:

```
6073 \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{leftedge}}%
```

Add a custom style if a vertical bar was requested:

```
6074 \ifdefstring{\LWR@tempone}{tvertbarl}{%
6075 \LWR@tdaddstyle%
6076 border-left: 1px solid \#\LWR@vertruleHTMLcolor%
6077 }{}%
6078 }{}%
```

Possibly add a right vertical rule for this cell:

```
6079 \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{\arabic{LWR@tablecolindex}}}%
6080 \ifdefstring{\LWR@tempone}{tvertbarr}{%
```

Add a custom style if a vertical bar was requested:

```
6081 \LWR@tdaddstyle%
6082 border-right: 1px solid \#\LWR@vertruleHTMLcolor%
6083 }{}%
6084 }%
6085 }
```

Ctrl LWR@cellcolordepth Counts how many cell color <div>s were added to the current tabular data cell.

```
6086 \newcounter{LWR@cellcolordepth}
```

\LWR@subaddtabularcellcolor *{\HTML color}*

```
6087 \newcommand*{\LWR@subaddtabularcellcolor}[1]{%
6088 \LWR@htmltag{div class="cellcolor" style="%
6089 background:\#{ }#1 %
6090 " }%
6091 \addtocounter{LWR@cellcolordepth}{1}%
6092 }
```

\LWR@addtabularcellcolor Adds a cell color style, if needed.

```
6093 \newcommand*{\LWR@addtabularcellcolor}{%
6094 \ifdefvoid{\LWR@cellHTMLcolor}%
```

```

6095 {%
6096     \ifdefvoid{\LWR@rowHTMLcolor}%
6097     {%
6098         \ifdefvoid{\LWR@xcolorrowHTMLcolor}%
6099         {%
6100             \ifdefvoid{\LWR@columnHTMLcolor}%
6101             {%
6102                 {\LWR@subadddtabularcellcolor{\LWR@columnHTMLcolor}}}%
6103             }%
6104             {\LWR@subadddtabularcellcolor{\LWR@xcolorrowHTMLcolor}}}%
6105         }%
6106         {\LWR@subadddtabularcellcolor{\LWR@rowHTMLcolor}}}%
6107     }%
6108 {\LWR@subadddtabularcellcolor{\LWR@cellHTMLcolor}}}%
6109 }

```

63.18 Multicolumns

63.18.1 Parsing multicolumns

```
6110 \newcounter{\LWR@tablemulticolwidth}
```

Indexes into the multicolumn specification:

```
6111 \newcounter{\LWR@tablemulticolspos}
```

Remembers multicolumn vertical rules if found in the column spec.

```
6112 \newbool{\LWR@mccolvertbarl}
```

```
6113 \newbool{\LWR@mccolvertbarr}
```

`\LWR@printmccoltype` `{<colspec>}` Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag's class.

```

6114 \newcommand*{\LWR@printmccoltype}[1]{%
6115 \LWR@traceinfo{\LWR@printmccoltype -#1-}%

```

Get one token of the column spec:

```
6116 \StrChar{#1}{\arabic{\LWR@tablemulticolspos}}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```

6117 \IfStrEq{\LWR@strresult}{l}{l}{}%
6118 \IfStrEq{\LWR@strresult}{c}{c}{}%
6119 \IfStrEq{\LWR@strresult}{r}{r}{}%
6120 \IfStrEq{\LWR@strresult}{p}{p}{}%
6121 \IfStrEq{\LWR@strresult}{m}{m}{}%
6122 \IfStrEq{\LWR@strresult}{b}{b}{}%
6123 \IfStrEq{\LWR@strresult}{P}{P}{}%
6124 \IfStrEq{\LWR@strresult}{M}{M}{}%
6125 \IfStrEq{\LWR@strresult}{B}{B}{}%
6126 \IfStrEq{\LWR@strresult}{S}{r}{}%
6127 \IfStrEq{\LWR@strresult}{X}{p}{}%

6128 \IfStrEq{\LWR@strresult}{|}{}%
6129     \ifnumcomp{value{\LWR@tablemulticolspos}}{=}{1}% left edge?
6130         {\booltrue{\LWR@mcolvertbarl}}% left edge
6131         {\booltrue{\LWR@mcolvertbarr}}% not left edge
6132 }{}%
6133 \LWR@traceinfo{lwarp@printmccoltype done}%
6134 }

```

`\LWR@multicolpartext` Print the data with paragraph tags:

```

6135 \newcommand*{\LWR@multicolpartext}{%
6136 \LWR@startpars%
6137 \LWR@multicoltext%
6138 \LWR@stoppars%
6139 }

```

`\LWR@multicolother` `{\colspec}` For @, !, >, <, print the next token without paragraph tags:

```

6140 \newcommand*{\LWR@multicolother}[1]{%
6141 \addtocounter{\LWR@tablemulticolspos}{1}%
6142 \StrChar{#1}{\arabic{\LWR@tablemulticolspos}}[\LWR@strresult]%
6143 \LWR@strresult%

```

A valid column data type was found:

```

6144 \booltrue{\LWR@validtablecol}%
6145 }

```

`\LWR@multicolskip` Nothing to print for this column type.

```

6146 \newcommand*{\LWR@multicolskip}{%

```

A valid column data type was found:

```
6147 \booltrue{LWR@validtablecol}%
6148 }
```

\LWR@printmccoldata {<colspec>} Print the data for any valid column type found.

```
6149 \newcommand*{\LWR@printmccoldata}[1]{%
6150 \LWR@traceinfo{lwr@printmccoldata -#1}%
```

Not yet found a valid column type:

```
6151 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
6152 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
6153 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
6154 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
6155 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
6156 \IfStrEq{\LWR@strresult}{D}{}%
6157 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
6158 \LWR@multicoltext%
6159 }{}%
6160 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
6161 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
6162 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
6163 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
6164 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
6165 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
6166 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
6167 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
6168 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
6169 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
6170 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
6171 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
6172 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%
```

If an invalid column type:

```
6173 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%
```

Tracing:

```
6174 \LWR@traceinfo{lwr@printmccoldata done}%
6175 }
```

```
\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.

```
6176 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
6177 \setcounter{\LWR@tablemulticolspos}{1}%
6178 \StrLen{#1}[\LWR@strresult]%
6179 \setcounter{\LWR@tablemulticolwidth}{\LWR@strresult}%
```

Scan across the tokens in the column spec:

```
6180 \whileboolexpr{%
6181     not test {%
6182         \ifnumcomp{\value{\LWR@tablemulticolspos}}{>}{\value{\LWR@tablemulticolwidth}}{%
6183             }
6184 }%
6185 {%
```

Execute the assigned print function for each token in the column spec:

```
6186 #2{#1}%
```

Move to the next token in the column spec:

```
6187 \addtocounter{\LWR@tablemulticolspos}{1}%
6188 }%
6189 }
```

63.18.2 Multicolumn factored code

```
\LWR@addmulticolvertrulecolor
```

```
6190 \newcommand*{\LWR@addmulticolvertrulecolor}{%
```

No vertical rules if finishing the tabular with a row of empty cells:

```
6191 \ifbool{\LWR@tabularmutemods}{-}{%
```

Left side:

```
6192     \ifbool{\LWR@mcolvertbarl}{%
6193         \LWR@tdaddstyle%
6194         border-left: 1px solid \#\LWR@vertruleHTMLcolor%
6195     }{}%
```

Right side:

```
6196 \ifbool{LWR@mcolvertbarr}{%
6197 \LWR@tdaddstyle%
6198 border-right: 1px solid \#\LWR@vertruleHTMLcolor%
6199 }{}%
6200 }%
6201 }
```

```
6202 \newcommand{\LWR@multicoltext}{}%
```

To find multicolumn right trim:

```
6203 \newcounter{LWR@lastmulticolumn}
```

```
\LWR@domulticolumn [<1: vpos>] [<2: #rows>] [<3: numLaTeXcols>] [<4: numHTMLcols>] [<5: colspec>]
[<6: text>]
```

```
6204 \NewDocumentCommand{\LWR@domulticolumn}{o o m m +m}{%
6205 \LWR@traceinfo{LWR@domulticolumn -#1- -#2- -#4- -#5-}%
```

Remember the text to be inserted, and remember that a valid column type was found:

```
6206 \renewcommand{\LWR@multicoltext}{%
6207 #6%
6208 \booltrue{LWR@validtablecol}%
6209 }%
```

Compute the rightmost column to be included. This is used to create the right trim.

```
6210 \setcounter{LWR@lastmulticolumn}{\value{LWR@tablecolindex}}%
6211 \addtocounter{LWR@lastmulticolumn}{#3}%
6212 \addtocounter{LWR@lastmulticolumn}{-1}%
```

Row processing:

```
6213 \LWR@maybenewtablerow%
```

Begin the opening table data tag:

```
6214 \LWR@htmltag{td colspan="#4" %
6215 \IfValueT{#2}{% rows?
6216 rowspan="#2" %
```

```

6217 \IfValueT{#1}{% vpos?
6218 \ifstrequal{#1}{b}{style="\LWR@origmbox{vertical-align:bottom}" }{}%
6219 \ifstrequal{#1}{t}{style="\LWR@origmbox{vertical-align:top}" }{}%
6220 }% vpos?
6221 }% rows?

```

```

6222 class="td%

```

Print the column type and vertical bars:

```

6223 \boolfalse{LWR@mcolvertbarl}%
6224 \boolfalse{LWR@mcolvertbarr}%
6225 \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:

```

6226 \LWR@subaddcmidruletrim%
6227 {\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
6228 {\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%

```

Also add vertical bar class.

```

6229 \ifbool{LWR@mcolvertbarl}{ tvertbarl}{}%
6230 \ifbool{LWR@mcolvertbarr}{ tvertbarr}{}%

```

Close the class tag’s opening quote:

```

6231 "%

```

```

6232 \LWR@tdstartstyles%

```

```

6233 \LWR@addtabulararrowcolor%

```

```

6234 \LWR@addcmidrulewidth%
6235 \LWR@addtabularhrulecolor%
6236 \LWR@addmulticolvertrulecolor%
6237 \LWR@addformatwppalignment{#5}%
6238 \LWR@tdendstyles%
6239 }% end of the opening table data tag
6240 \global\boolfalse{LWR@intabularmetadata}%
6241 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
6242 }

```

63.18.3 Multicolumn

`\LWR@htmlmulticolumn` $\{\langle numcols \rangle\} \{\langle alignment \rangle\} \{\langle text \rangle\}$

```
6243 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
6244 {%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
6245 \LWR@tabularhtmlcolumns{\arabic{\LWR@tablecolindex}}{\#1}
```

Create the multicolumn tag:

```
6246 \LWR@domulticolumn{\#1}{\arabic{\LWR@tabhtmlcoltotal}}{\#2}{\#3}%
```

Move to the next \LaTeX column:

```
6247 \addtocounter{\LWR@tablecolindex}{\#1}%
6248 \addtocounter{\LWR@tablecolindex}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
6249 \booltrue{\LWR@skipatbang}%
6250 }
```

63.18.4 Longtable captions

longtable captions use `\multicolumn`.

`Bool` `LWR@starredlongtable` Per the caption package, step the counter if longtable*.

```
6251 \newbool{\LWR@starredlongtable}
6252 \boolfalse{\LWR@starredlongtable}
```

Per the caption package. User-redefinable float type.

```
6253 \providecommand*\LTcapttype{table}
```

`\LWR@longtabledatacaptiontag` * [*toc entry*] $\{\langle caption \rangle\}$

```
6254 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
6255 {%
```

Remember the latest name for `\nameref`:

```

6256 \IfValueTF{#2}{% optional given?
6257     \ifblank{#2}% optional empty?
6258     {\LWR@setlatestname{#3}}% empty
6259     {\LWR@setlatestname{#2}}% given and non-empty
6260 }% optional given
6261 {\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:

```

6262 \LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalcols}}

```

Create the multicolumn tag:

```

6263 \LWR@domulticolumn{\arabic{LWR@tabletotalcols}}{\arabic{LWR@tabhtmlcoltotal}}{P}%
6264 {% \LWR@domulticolumn
6265 \IfBooleanTF{#1}% star?

```

Star version, show a caption but do not make a LOT entry:

```

6266 {% yes star
6267     \LWR@figcaption%
6268     #3%
6269     \endLWR@figcaption%
6270 }%
6271 {% No star:

```

Not the star version:

Don't step the counter if \caption[] {A caption.}

```

6272     \ifbool{LWR@starredlongtable}%
6273     {%
6274         \ifblank{#2}% TOC entry
6275         {%
6276             {%
6277                 \refstepcounter{\LTcapttype}%
6278                 \protected@edef\currentlabel{%
6279                     \csuse{p@\LTcapttype}\csuse{the\LTcapttype}}%
6280             }%
6281         }{%

```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

6282     \LWR@figcaption%
6283     \csuse{fnum@\LTcapttype}\CaptionSeparator#3%
6284     \endLWR@figcaption%

```

See if an optional caption was given:

```
6285 \ifblank{#2}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
6286 {}%
```

If the optional caption was given, but might only be []:

```
6287 {% TOC entry not empty
6288 \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```
6289 {% No TOC entry
6290 \addcontentsline%
6291 {\csuse{ext@LTcapttype}}%
6292 {LTcapttype}%
6293 {%
6294 \protect\numberline%
6295 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
6296 {\ignorespaces #3\protect\relax}%
6297 }%
6298 }% end of No TOC entry
```

The optional caption has text enclosed:

```
6299 {% yes TOC entry
6300 \addcontentsline%
6301 {\csuse{ext@LTcapttype}}%
6302 {LTcapttype}%
6303 {%
6304 \protect\numberline%
6305 {\csuse{p@LTcapttype}\csuse{theLTcapttype}}%
6306 {\ignorespaces #2\protect\relax}%
6307 }%
6308 }% end of yes TOC entry
6309 }% end of TOC entry not empty
6310 }% end of no star
```

Skip any trailing @ or ! columns for this cell:

```
6311 \booltrue{LWR@skipatbang}%
6312 }% end of \LWR@domulticolumn
6313
6314 \addtocounter{LWR@tablecolindex}{\arabic{LWR@tabletotalcols}}
6315 \addtocounter{LWR@tablecolindex}{-1}
```

```
6316
6317 }
```

63.18.5 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.

```
6318 \newcounter{LWR@tabhtmlcolindex}
6319 \newcounter{LWR@tabhtmlcolend}
6320 \newcounter{LWR@tabhtmlcoltotal}
```

```
\LWR@subtabularhtmlcolumns {<index>}
```

Factored from `\LWR@tabularhtmlcolumns`, which follows.

```
6321 \newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the `@` specification for this column:

```
6322     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the `@` specification is not empty, add to the count:

```
6323     \ifdefempty{\LWR@atbangspec}%
6324         {}%
6325         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the `!` columns:

```
6326     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{#1}}%
6327     \ifdefempty{\LWR@atbangspec}%
6328         {}%
6329         {\addtocounter{LWR@tabhtmlcoltotal}{1}}%
6330 }
```

```
\LWR@tabularhtmlcolumns {<starting  $\text{\TeX}$  column>} {<number  $\text{\TeX}$  columns>}
```

Compute the total number of HTML columns being spanned, considering the starting \LaTeX table column and the number of \LaTeX tabular columns being spanned. Any @ and ! columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter LWR@tabhtmlcoltotal.

```
6331 \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:

```
6332 \setcounter{LWR@tabhtmlcolindex}{#1}%
6333 \setcounter{LWR@tabhtmlcoltotal}{#2}%
6334 \setcounter{LWR@tabhtmlcolend}{#1}%
6335 \addtocounter{LWR@tabhtmlcolend}{#2}%
```

If at the left edge, add the at/bang columns for the left edge:

```
6336 \ifnumcomp{\value{LWR@tabhtmlcolindex}}{=}{1}{%
6337     \LWR@subtabularhtmlcolumns{leftedge}%
6338 }{ }%
```

Walk across the \LaTeX columns looking for @ and ! columns:

```
6339 \whileboolexpr{%
6340     test {%
6341         \ifnumcomp{\value{LWR@tabhtmlcolindex}}{<}{\value{LWR@tabhtmlcolend}}}%
6342     }%
6343 }%
6344 {%
6345     \LWR@subtabularhtmlcolumns{\arabic{LWR@tabhtmlcolindex}}%
6346     \addtocounter{LWR@tabhtmlcolindex}{1}%
6347 }% \whiledo
6348 }

6349 \end{warpHTML}
```

63.19 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a lateximage.

See section 225 for the HTML versions.

for HTML & PRINT: 6350 \begin{warpall}

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:
[https://tex.stackexchange.com/questions/168434/](https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex)
[problem-with-abbreviation-of-multirow-and-multicolumn-latex](https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex)

After the user may have

```
6351 \AtBeginDocument{
```

`\@ifundefined{@xmultirow}` determines if multirow was never loaded.

```
6352 \@ifundefined{@xmultirow}
6353 {}% no version of multirow was loaded
6354 {% \@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.

```
6355 \@ifpackageloaded{multirow}{% v2.0 or newer
6356 \@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
6357 {% v2.0+:
6358 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
6359     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
6360 {\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}[#5]{#6}[#7]{#8}}}%
6361 }
6362 {% loaded but older, probably not executed:
6363 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
6364     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
6365 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
6366 }
6367 }% 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.

```
6368 {% v1.6 or older did not \ProvidePackage
6369 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
6370     {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
6371 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5]{#6}[#7]{#8}}}%
6372 }
```

The user-level interface. This is provided if the HTML version was not already given.

```
6373 \providecommand*{\multicolumnrow}{\LWR@origmulticolumnrow}
```

```

6374 }% \@xmultirow defined, so multirow was loaded
6375 }% AtBeginDocument

6376 \end{warpall}

```

63.20 Utility macros inside a table

for HTML output: 6377 \begin{warpHTML}

Used to prevent opening a tabular data cell if the following token is one which does not create tabular data:

```
6378 \newcommand*{\LWR@donothing}{}

```

In case bigdelim is not loaded:

```

6379 \newcommand*{\ldelim}{}
6380 \newcommand*{\rdelim}{}

```

```
6381 \end{warpHTML}

```

63.21 Special-case tabular markers

for HTML & PRINT: 6382 \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 8.8.


```
6383 \newcommand*{\TabularMacro}{}

```

```
6384 \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: 6385 \begin{warpHTML}

```

6386 \newcommand*{\ResumeTabular}{%
6387 \global\boolfalse{LWR@exitingtabular}%
6388 \global\boolfalse{LWR@tabularmutemods}%
6389 \LWR@getmynexttoken%
6390 }

```

```

6391 \end{warpHTML}

```

for PRINT output: 6392 \begin{warpprint}

```

6393 \newcommand*{\ResumeTabular}{}
6394 \end{warpprint}

```

63.22 Checking for a new table cell

for HTML output: 6395 \begin{warpHTML}

Bool LWR@exitingtabular When \end is found, turns off the next opening data tag.

```

6396 \newbool{LWR@exitingtabular}

```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```

6397 \newbool{LWR@tabularmutemods}

```

`\LWR@tabledatacolumnntag` Open a new HTML table cell unless the next token is for a macro which does not create data, such as `\hline`, `\toprule`, etc:

```
6398 \newcommand*{\LWR@tabledatacolumnntag}%
6399 {%
6400 \LWR@traceinfo{\LWR@tabledatacolumnntag}%
```

`\show\LWR@mynexttoken` to see what tokens to look for

If not any of the below, start a new table cell:

```
6401 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
6402 \ifdefequal{\LWR@mynexttoken}{\end}%
6403     {\global\booltrue{\LWR@exitingtabular}}{}%
```

`longtable` can have a caption in a cell

```
6404 \ifdefequal{\LWR@mynexttoken}{\caption}%
6405     {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
6406 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
6407     {\let\mynext\LWR@donothing}{}%
6408 \ifdefequal{\LWR@mynexttoken}{\multirow}%
6409     {\let\mynext\LWR@donothing}{}%
6410 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
6411     {\let\mynext\LWR@donothing}{}%
6412 \ifdefequal{\LWR@mynexttoken}{\noalign}%
6413     {\let\mynext\LWR@donothing}{}%
```

If an `\mrowcell`, this is a cell to be skipped over:

```
6414 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
6415     {\let\mynext\LWR@donothing}{}%
```

If an `\mcolrowcell`, this is a cell to be skipped over:

```
6416 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
6417     {\let\mynext\LWR@donothing}{}%
6418 %
6419 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
6420     {\let\mynext\LWR@donothing}{}%
```

```
6421 %
6422 \ifdefequal{\LWR@mynexttoken}{\hline}%
6423     {\let\mynext\LWR@donothing}{}%
6424 %
6425 \ifdefequal{\LWR@mynexttoken}{\firsthline}%
6426     {\let\mynext\LWR@donothing}{}%
6427 %
6428 \ifdefequal{\LWR@mynexttoken}{\lasthline}%
6429     {\let\mynext\LWR@donothing}{}%
6430 %
6431 \ifdefequal{\LWR@mynexttoken}{\toprule}%
6432     {\let\mynext\LWR@donothing}{}%
6433 %
6434 \ifdefequal{\LWR@mynexttoken}{\midrule}%
6435     {\let\mynext\LWR@donothing}{}%
6436 %
6437 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
6438     {\let\mynext\LWR@donothing}{}%
6439 %
6440 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
6441     {\let\mynext\LWR@donothing}{}%
6442 %
6443 \ifdefequal{\LWR@mynexttoken}{\cline}%
6444     {\let\mynext\LWR@donothing}{}%
6445 %
6446 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
6447     {\let\mynext\LWR@donothing}{}%
6448 %
6449 \ifdefequal{\LWR@mynexttoken}{\rowcolor}%
6450     {\let\mynext\LWR@donothing}{}%
6451 %
6452 \ifdefequal{\LWR@mynexttoken}{\arrayrulecolor}%
6453     {\let\mynext\LWR@donothing}{}%
6454 %
6455 \ifdefequal{\LWR@mynexttoken}{\doublerulesepcolor}%
6456     {\let\mynext\LWR@donothing}{}%
6457 %
6458 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
6459     {\let\mynext\LWR@donothing}{}%
6460 %
6461 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
6462     {\let\mynext\LWR@donothing}{}%
6463 %
6464 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
6465     {\let\mynext\LWR@donothing}{}%
6466 %
6467 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
6468     {\let\mynext\LWR@donothing}{}%
```

Ignore an empty line between rows:

```
6469 \ifdefequal{\LWR@mynexttoken}{\par}%
6470     {\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:

```
6471 \LWR@traceinfo{\LWR@tabledatacolumnstag: about to do mynext}%
6472 \mynext%
6473 \LWR@traceinfo{\LWR@tabledatacolumnstag: done}%
6474 }

6475 \end{warpHTML}
```

63.23 `\mrowcell`

for HTML & PRINT: 6476 `\begin{warpall}`


`\mrowcell` The user must insert `\mrowcell` into any `\multirow` cells which must be skipped.  This command has no action during print output.

```
6477 \newcommand*{\mrowcell}{}

6478 \end{warpall}
```

63.24 `\mcolrowcell`

for HTML & PRINT: 6479 `\begin{warpall}`

`\mcolrowcell` The user must insert `\mcolrowcell` into any `\multicolumnrow` cells which must be skipped.  This command has no action during print output.

```
6480 \newcommand*{\mcolrowcell}{}

6481 \end{warpall}
```

63.25 New tabular environment

for HTML output: 6482 `\begin{warpHTML}`

These are default definitions in case booktabs is not loaded, and are not expected to be used, but must exist as placeholders.

```
6483 \newcommand*{\LWR@origtoprule}[1] [] {\hline}
6484 \newcommand*{\LWR@origmidrule}[1] [] {\hline}
6485 \LetLtxMacro\LWR@origcmidrule\cline
6486 \newcommand*{\LWR@origbottomrule}[1] [] {\hline}
6487 \newcommand*{\LWR@origaddlinespace}[1] [] {}
6488 \newcommand*{\LWR@origmorecmidrules}{}
6489 \newcommand*{\LWR@origspecialrule}[3] {\hline}
```

`\noalign` `{\text{}}` Redefined for use inside tabular.

```
6490 \LetLtxMacro\LWR@orignoalign\noalign
6491
6492 \newcommand{\LWR@tabularnoalign}[1] {%
6493 \begingroup%
6494 \global\advance\rownum\m@ne%
6495 \renewcommand*{\LWR@xcolorrowHTMLcolor}{}%
6496 \multicolumn{\value{\LWR@tabletotalcols}}{1}{#1} \\
6497 \endgroup%
6498 % \@rowcolors%
6499 \LWR@getmynexttoken%
6500 }
```

`\LWR@HTMLhline` The definition of `\hline` depends on whether `tabls` has been loaded. If so, optional space below the line may be specified, but will be ignored.

```
6501 \AtBeginDocument{
6502 \ifpackageloaded{lwarp-tbls}
6503 {
6504 \newcommand*{\LWR@HTMLhline}[1] [] {%
6505 \ifbool{FormatWP}%
6506 {\LWR@docmidrule{1-\arabic{\LWR@tabletotalcols}}}%
6507 {\booltrue{\LWR@doinghline}}%
6508 \LWR@getmynexttoken%
6509 }
6510 {
6511 \newcommand*{\LWR@HTMLhline} {%
6512 \ifbool{FormatWP}%
6513 {\LWR@docmidrule{1-\arabic{\LWR@tabletotalcols}}}%
6514 {\booltrue{\LWR@doinghline}}%
6515 \LWR@getmynexttoken%
6516 }
```

```
6516 }
6517 }% AtBeginDocument
```

`\LWR@HTMLcline` $\{\langle columns \rangle\}$

```
6518 \NewDocumentCommand{\LWR@HTMLcline}{m}%
6519 {\LWR@docmidrule{#1}\LWR@getmynexttoken}%
```

`\LWR@nullifyNoAutoSpacing` For babel-french, turn off auto spacing at the start of the tabular, then nullify the autospacing commands inside the tabular, since they were not compatible with the tabular column parsing code, which uses xstring.

```
6520 \AtBeginDocument{
6521 \ifundefined{frenchbsetup}%
6522 {% no babel-french
6523   \newcommand*{\LWR@nullifyNoAutoSpacing}{}
6524 }% no babel-french
6525 {% yes babel-french
6526   \newcommand*{\LWR@nullifyNoAutoSpacing}{%
6527     \NoAutoSpacing%
6528     \renewcommand*{\NoAutoSpacing}{}%
6529     \renewcommand*{\LWR@FBcancel}{}%
6530   }
6531 }% yes babel-french
6532 }% AtBeginDocument
```

Env `LWR@tabular` $[\langle vertposition \rangle] \{\langle colspecs \rangle\}$

The new tabular environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```
6533 \StartDefiningTabulars
6534
6535 \newenvironment*{LWR@tabular}[2] []
6536 {%
6537 \LWR@traceinfo{LWR@tabular started}%
6538 \addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
6539 \global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```
6540 \global\boolfalse{LWR@doinghline}%
6541 \global\boolfalse{LWR@doingtbrule}%
6542 \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.

```
6543 \LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the @ and ! columns.

```
6544 \global\boolfalse{LWR@exitingtabular}%
6545 \global\boolfalse{LWR@tabularmutemods}%
```

Create the table tag:

```
6546 \global\booltrue{LWR@intabularmetadata}%
6547 \LWR@forcenewpage
6548 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
6549 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
6550 \LWR@stoppars%
```

Track column #:

```
6551 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
6552 \boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
6553 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
6554 \LetLtxMacro{\\}{\LWR@tabularendoffline}%
```

The following adjust for colortbl:

```
6555 \LetLtxMacro\columncolor\LWR@HTMLcolumncolor%
6556 \LetLtxMacro\rowcolor\LWR@HTMLrowcolor%
6557 \LetLtxMacro\cellcolor\LWR@HTMLcellcolor%
```

```

6558 \LetLtxMacro\arrayrulecolor\LWR@HTMLarrayrulecolor%
6559 \LetLtxMacro\doublerulesepcolor\LWR@HTMLdoublerulesepcolor%
6560 \renewcommand*{\LWR@columnHTMLcolor}{}%
6561 \renewcommand*{\LWR@rowHTMLcolor}{}%
6562 \renewcommand*{\LWR@cellHTMLcolor}{}%
6563 \@rowcolors%

```

The vertical rules are set to the color active at the start of the tabular. `\arrayrulecolor` will then affect horizontal rules inside the tabular, but not the vertical rules.

```

6564 \edef\LWR@vertruleHTMLcolor{\LWR@ruleHTMLcolor}%

```

Tracking the depth of cell color <div>s:

```

6565 \setcounter{LWR@cellcolordepth}{0}%

```

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 `tabs` package.

```

6566 \LWR@traceinfo{LWR@tabular: redefining macros}%
6567 \LetLtxMacro\noalign\LWR@tabularnoalign%
6568 \LetLtxMacro\hline\LWR@HTMLhline%
6569 \LetLtxMacro\cline\LWR@HTMLcline%

6570 \DeclareDocumentCommand{\toprule}{o d()}%
6571   {%
6572     \IfValueTF{##1}%
6573       {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6574       {%
6575         \ifbool{FormatWP}%
6576         {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6577         {\booltrue{LWR@doingtbrule}}}%
6578       }%
6579   \LWR@getmynexttoken}%
6580 %
6581 \DeclareDocumentCommand{\midrule}{o d()}%
6582   {%
6583     \IfValueTF{##1}%
6584       {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6585       {%
6586         \ifbool{FormatWP}%
6587         {\LWR@docmidrule[##1](){1-\arabic{LWR@tabletotalcols}}}%
6588         {\booltrue{LWR@doinghline}}}%
6589       }%
6590   \LWR@getmynexttoken}%

```

```

6591 %
6592 \DeclareDocumentCommand{\cmidrule}{0{\LWR@cmidrulewidth} d() m}%
6593 {\LWR@docmidrule[##1](##2){##3}\LWR@getmynexttoken}%
6594 %
6595 \DeclareDocumentCommand{\bottomrule}{o d()}%
6596   {%
6597     \IfValueTF{##1}%
6598       {\LWR@docmidrule[##1](){1-\arabic{\LWR@tabletotalcols}}}%
6599       {%
6600         \ifbool{FormatWP}%
6601           {\LWR@docmidrule[##1](){1-\arabic{\LWR@tabletotalcols}}}%
6602           {\booltrue{\LWR@doingtbrule}}}%
6603       }%
6604   \LWR@getmynexttoken}%
6605 %
6606 \DeclareDocumentCommand{\addlinespace}{o}{}%
6607 \DeclareDocumentCommand{\morecmidrules}{l}{}%
6608 \DeclareDocumentCommand{\specialrule}{m m m d()}%
6609   {\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.

```

6610 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
6611 \renewcommand*{\mrowcell}{%
6612   \LWR@maybenewtablerow%
6613   \LWR@tabularleftedge%
6614   \global\booltrue{\LWR@skippingmrowcell}%
6615 }%
6616 \renewcommand*{\mcolrowcell}{%
6617   \LWR@maybenewtablerow%
6618   \global\booltrue{\LWR@skippingmcolrowcell}%
6619 }%
6620 \LetLtxMacro\caption\LWR@longtabledatacaptiontag%

```

Reset for new processing:

```

6621 \global\boolfalse{\LWR@tableparcell}%
6622 \global\boolfalse{\LWR@skippingmrowcell}%
6623 \global\boolfalse{\LWR@skippingmcolrowcell}%
6624 \global\boolfalse{\LWR@skipatbang}%
6625 \global\boolfalse{\LWR@emptyatbang}%

```

Set & for its special meaning inside the tabular:

```

6626 \StartDefiningTabulars%
6627 \protected\gdef&{\LWR@tabularampersand}%

```

Nest one level deeper of tabular paragraph handling:

```
6628 \addtocounter{LWR@tabularpardepth}{1}%
```

Look ahead for a possible table data cell:

```
6629 \LWR@traceinfo{LWR@tabular: about to LWR@getmynexttoken}%
6630 \LWR@getmynexttoken%
6631 }%
```

Ending the environment:

```
6632 {%
6633 \LWR@traceinfo{LWR@tabular ending}%
```

Unnest one level of tabular paragraph handling:

```
6634 \addtocounter{LWR@tabularpardepth}{-1}%
6635 \ifboolexpr{%
6636   test {%
6637     \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
6638   } or %
6639   (%
6640     bool{LWR@intabularmetadata} and%
6641     not bool{LWR@tabularcelladded} and%
6642     test {%
6643       \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}%
6644     }%
6645   )%
6646 }%
6647 {%
6648   \LWR@tabularfinishrow%
6649 }%
6650 {%
6651   \LWR@closetabledatacell%
6652 }%
6653 \LWR@htmlblocktag{/tr}%
```

xcolor row color support:

```
6654 \@rowcolor%

6655 \LWR@htmlblocktag{/table}%
6656 \global\boolfalse{LWR@intabularmetadata}%
```

Unnest one level of tabular:

```
6657 \addtocounter{LWR@tabulardepth}{-1}%
```

Restore & to its usual meaning:

```
6658 \protected\gdef&{\LWR@origampmacro}%
6659 \EndDefiningTabulars%
6660 \LWR@traceinfo{\LWR@tabular finished ending}%
6661 }
6662
6663 \EndDefiningTabulars

6664 \end{warpHTML}
```

64 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: 6665 \begin{warpHTML}

64.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
6666 \providecommand*{\@currentlabelname}{}%
```

`\LWR@stripperperiod` `{\langle text \rangle} [(\cdot)]`

Removes a trailing period.

```
6667 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` `{\langle object name \rangle}`

Removes `\label`, strips any final period, and remembers the result.

```
6668 \newcommand*{\LWR@setlatestname}[1]{%
```

Table 10: Cross-referencing data structures

Original \LaTeX:	(print and HTML)
<code>\refstepcounter:</code> Steps the counter and sets <code>\@currentlabel</code> . <code>\@currentlabel:</code> <code>\p@<ctr>\the<ctr></code> Updated by <code>\refstepcounter</code> . <code>\label:</code> Writes to the .aux file: <code>\newlabel{<label>}{\@currentlabel}{\thepage}}</code> <code>\newlabel:</code> When the .aux file is read, sets <code>\r@<label></code> . <code>\r@<label>:</code> Set to: <code>{\@currentlabel}{\thepage}}</code> <code>\ref:</code> Returns the first part of <code>\r@<label></code> . <code>\pageref:</code> Returns the second part of <code>\r@<label></code> .	
Added by lwarp:	(HTML only)
<code>\label:</code> Adds HTML tags (section 64.3), plus <code>\splabel</code> data (section 64.2): <code>zLWR@name:</code> The section name for this label. <code>zLWR@htmlfilenumber:</code> The file number or name for this label. <code>zLWR@lateximagedepth:</code> The <code>lateximagedepth</code> for this label. <code>zLWR@lateximagenumber:</code> The <code>lateximagenumber</code> for this label. <code>\nameref:</code> Emulated from <code>hyperref</code> for <code>lwarp</code> . See section 64.4. <code>\ref</code> and <code>\nameref:</code> Adds HTML tags. See section 64.4.	
Added by amsmath:	(print and HTML)
<code>\label:</code> Execution is delayed until the math environment is completed. <code>\ltx@label:</code> \LaTeX <code>\label</code> , (HTML: patched by <code>lwarp</code>), later patched by <code>cleveref</code> .	
Added by cleveref:	(print and HTML)
<code>\refstepcounter:</code> Added: sets <code>\cref@currentlabel</code> . <code>\cref@currentlabel:</code> (<code><type>=<ctr></code> unless an alias is used): <code>[<type>][\arabic{<ctr>}][<parent> <trs>]{\p@<ctr>\the<ctr>}</code> Also see section 50.4 for use with footnotes. <code>\label:</code> Writes to the .aux file: <code>\newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}</code> <code>\newlabel:</code> (Unchanged.) When the .aux file is read, sets <code>\r@<label>@cref</code> . <code>\r@<label>@cref:</code> Set to: <code>{\cref@currentlabel}{\thepage}}</code> Utility functions: See <code>\cref@getlabel</code> , <code>\cref@gettype</code> , <code>\cref@getcounter</code> , <code>\cref@getprefix</code> . Cross-referencing names: <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.	
Additionally patched by lwarp:	(HTML only)
<code>\cref, etc.:</code> Modified for <code>lwarp</code> . See section 77. <code>\label inside math:</code> See section 70.5.1.	
Footnotes: See <code>\noteentry</code> in section 50.4.	

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `getttitlestring`.

```
6669 \GetTitleStringExpand{#1}%
6670 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
6671 \edef\@currentlabelname{%
6672 \expandafter\LWR@stripperperiod\@currentlabelname%
6673 \ltx@empty.\ltx@empty\@nil%
6674 }%
6675 }
```

64.2 Zref setup

See:

[http://tex.stackexchange.com/questions/57194/
extract-section-number-from-equation-reference](http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference)

Create a new property list called `special`:

```
6676 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
6677 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
6678 \zref@newprop{zLWR@htmlfilenumber}{%
6679 \ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}%
6680 }%
```

Additional properties for `lateximages`:

```
6681 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
6682 \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`:

```
6683 \zref@addprop{special}{zLWR@name}
6684 \zref@addprop{special}{zLWR@htmlfilenumber}
6685 \zref@addprop{special}{zLWR@lateximagedepth}
6686 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
6687 \newcommand*{\LWR@sprof}[2]{%
6688 \zref@extractdefault{#1}{#2}{??}%
6689 }
```

`\LWR@nameref` $\{\langle label \rangle\}$ Returns the section name for this label:

```
6690 \newcommand*{\LWR@nameref}[1]{%
6691 \LWR@sprof{#1}{zLWR@name}%
6692 }
```

`\LWR@htmlfileref` $\{\langle label \rangle\}$ Returns the file number for this label:

```
6693 \newcommand*{\LWR@htmlfileref}[1]{%
```

DO NOT USE `\LWR@traceinfo` HERE! Will be expanded.

```
6694 \LWR@sprof{#1}{zLWR@htmlfilenumber}%
6695 }
```

`\LWR@lateximagedepthref` $\{\langle label \rangle\}$ Returns the `lateximagedepth` for this label:

```
6696 \newcommand*{\LWR@lateximagedepthref}[1]{%
6697 \LWR@sprof{#1}{zLWR@lateximagedepth}%
6698 }
```

`\LWR@lateximagenumberref` $\{\langle label \rangle\}$ Returns the `lateximagenumber` for this label:

```
6699 \newcommand*{\LWR@lateximagenumberref}[1]{%
6700 \LWR@sprof{#1}{zLWR@lateximagenumber}%
6701 }
```

`\LWR@splabel` $\{\langle label \rangle\}$ Sanitize the name and then creates the label:

```
6702 \newcommand*{\LWR@splabel}[1]{%
6703 \LWR@traceinfo{\LWR@splabel !#1!}%
6704 \LWR@setlatestname{\@currentlabelname}%
6705 \zref@labelbylist{#1}{special}%
6706 }
```

64.3 Labels

`\LWR@sublabel` $\{\langle label \rangle\}$ Creates an HTML id tag.

`\detokenize` is used to allow underscores in the labels.

```
6707 \newcommand*{\LWR@sublabel}[1]{%
6708 \LWR@traceinfo{\LWR@sublabel !#1!}%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
6709 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}{%
6710 {}}%
6711 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
6712 \LWR@sanitize{#1}%
6713 \ifbool{\LWR@doingstartpars}%
6714 {% pars allowed
6715 \ifbool{\LWR@doingapar}%
6716 {% par started
6717 \LWR@htmltag{a \LWR@origmbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
6718 }% par started
6719 {% par not started
6720 \LWR@stoppars%
6721 \LWR@htmltag{a \LWR@origmbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
6722 \LWR@startpars%
6723 }% par not started
6724 }% pars allowed
6725 {% pars not allowed
6726 \LWR@htmltag{a \LWR@origmbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
6727 }% pars not allowed
6728 }% not lateximage
6729 }
```

`\LWR@newlabel` (*<bookmark>*) [*<label>*] [*<type>*]

`\label` during HTML output when not in SVG math mode, removing extra spaces around the label, as done by regular \LaTeX `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional *<bookmark>* is per the memoir class, and is ignored.

The optional *<type>* is per the ntheorem package, and is ignored.

```
6730 \NewDocumentCommand{\LWR@newlabel}{d() m o}{%
6731 \LWR@traceinfo{\LWR@newlabel: starting}%
6732 \LWR@traceinfo{\LWR@newlabel: !#2!}%
6733 % \@bsphack%
```

Create a traditional \LaTeX label, as modified by cleveref:

```
6734 \LWR@origlabel{#2}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```
6735 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
6736 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
6737 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
6738 \LWR@splabel{#2}%
6739 \LWR@sublabel{#2}%
6740 % \@esphack%
6741 \LWR@traceinfo{LWR@newlabel: done}%
6742 }
```

64.4 References

`\LWR@startref` `{\label}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```
6743 \newcommand*{\LWR@startref}[1]
6744 {%
6745 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
6746 \LWR@sanitize{#1}%
6747 \LWR@traceinfo{LWR@startref A: !#1!}%
```

Create the filename part of the link:

```
6748 \LWR@htmltag{a href="%
6749 \LWR@traceinfo{LWR@startref B}%
6750 \LWR@origmbox{\LWR@htmlrefsectionfilename{#1}}%
6751 \LWR@traceinfo{LWR@startref C}%
6752 \#%
```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```
6753 \LWR@traceinfo{LWR@startref D: !#1!}%
6754 \ifthenelse{\equal{\LWR@lidref}{??}}%
```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

6755 {%
6756   \LWR@traceinfo{LWR@startref D0: ??}%
6757   ??%
6758 }%

```

If LWR@lateximagedepth is known. Use a lateximage if the depth is greater than zero, or a regular link otherwise:

```

6759 {%
6760   \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
6761   \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
6762   {%
6763     \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
6764     lateximage\LWR@lateximagenumberref{#1}%
6765   }%
6766   {%
6767     \LWR@traceinfo{LWR@startref D3}%

```

\detokenize is used to allow underscores in the labels:

```

6768   \LWR@origmbox{\LWR@sanitized}%
6769   }%
6770 }%
6771 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

6772 "%
6773 \LWR@traceinfo{LWR@startref F}%
6774 }

```

`\LWR@subnewref` `{\langle label \rangle}` `{\langle label or sub@label \rangle}`

Factored for the subfig package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```

6775 \NewDocumentCommand{\LWR@subnewref}{m m}{%
6776   \LWR@traceinfo{LWR@subnewref #1 #2}%
6777   \LWR@startref{#1}%
6778   \LWR@origref{#2}%
6779   \LWR@htmltag{/a}%
6780 }

```

`\ref` * `{\langle label \rangle}` `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` * $\langle label \rangle$ Create an internal document reference link, or without a link if starred per hyperref.

```
6781 \NewDocumentCommand{\LWR@newref}{s m}{%
6782 \LWR@traceinfo{\LWR@newref !#2!}%
6783 \IfBooleanTF{#1}%
6784 {\LWR@origref{#2}}%
6785 {\LWR@subnewref{#2}{#2}}%
6786 }
```

`\pagerefPageFor` Text for page references.

```
6787 \newcommand*{\pagerefPageFor}{see }
```

`\pageref` * $\langle label \rangle$ Create an internal document reference, or just the unlinked number if starred, per hyperref.

```
6788 \NewDocumentCommand{\LWR@newpageref}{s m}{%
6789 \IfBooleanTF{#1}%
6790 {(\pagerefPageFor\LWR@origref{#2})}%
6791 {(\cpageref{#2})}%
6792 }
```

`\nameref` $\langle label \rangle$

```
6793 \newrobustcmd*{\nameref}[1]{%
6794 \LWR@traceinfo{nameref}%
6795 \LWR@startref{#1}%
6796 \LWR@traceinfo{nameref B}%
6797 \LWR@nameref{#1}%
6798 \LWR@traceinfo{nameref C}%
6799 \LWR@htmltag{/a}%
6800 \LWR@traceinfo{nameref: done}%
6801 }
```

`\Nameref` $\langle label \rangle$ In print, adds the page number. In HTML, does not.

```
6802 \LetLtxMacro\Nameref\nameref
```

64.5 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.

```
6803 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
6804 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Emulates `hyperref`:

`\@currentHref` Added to support `backref`.

```
6805 \AtBeginDocument{
6806 \def\@currentHref{%
6807 autopage-\theLWR@currentautosec%
6808 }
6809 }
```

Create a link with a text name:

`\LWR@subhyperref` `{\langle URL \rangle}{\langle text \rangle}`

```
6810 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
6811 \LWR@traceinfo{\LWR@subhyperref !#1!}%
6812 \LWR@sanitize{#1}%
6813 \LWR@htmltag{%
6814   a href="\LWR@sanitized" %
6815 %   a href="%
6816 %   \begingroup\@sanitize#1\endgroup%
6817 %   " %
6818   target="_{}blank"\LWR@orignewline%
6819 }%
6820 #2%
6821 \LWR@htmltag{/a}%
6822 \LWR@ensuredoingapar%
6823 }
```

`\LWR@subhyperrefclass` `{\langle URL \rangle}{\langle text \rangle}{\langle htmlclass \rangle}`

```
6824 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
```

```

6825 % \LWR@sanitize{#1}%
6826 \LWR@htmltag{%
6827 %     a href="\LWR@sanitized"
6828 %     a href="%
6829 %         \begingroup\@sanitize#1\endgroup%
6830 %     " %
6831 %     class="#3"\LWR@orignewline%
6832 }%
6833 #2%
6834 \LWR@htmltag{/a}%
6835 \LWR@ensuredoingapar%
6836 }

```

`\href` [*<options>*] {*<URL>*} {*<text>*}

Create a link with accompanying text:

```

6837 \DeclareDocumentCommand{\href}{0{ } m +m}{%
6838 \LWR@ensuredoingapar%
6839 \LWR@subhyperref{#2}{#3}%
6840 }

```

`\nolinkurl` {*<URL>*}

Print the name of the link without creating the link:

```

6841 \newcommand*{\nolinkurl}[1]{%
6842 \LWR@ensuredoingapar%
6843 \def\LWR@templink{#1}%
6844 \@onelevel@sanitize\LWR@templink%
6845 \LWR@templink%
6846 }

```

`\url` {*<URL>*}

Create a link whose text name is the address of the link. The url package may redefine `\url`, so it is `\let` to `\LWR@urlahere` and also redefined by `lwarp-url`.

```

6847 \DeclareDocumentCommand{\url}{m}{%
6848 \LWR@ensuredoingapar%
6849 \def\LWR@templink{#1}%
6850 \@onelevel@sanitize\LWR@templink%
6851 \href{\LWR@templink}{\LWR@templink}%
6852 }

```

`\LWR@subinlineimage` [*<alttag>*] {*<class>*} {*<filename>*} {*<extension>*} {*<style>*}

```
6853 \newcommand*{\LWR@subinlineimage}[5] [] {%  
6854 \ifblank{#1}%  
6855 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}%  
6856 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"}}%  
6857 }  
  
6858 \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>` “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.
-

65 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”.

65.1 Float captions

for HTML output: 6859 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>]

Begins a \newfloat environment.

```
6860 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
6861 \ifbool{FormatWP}{\newline}{}%
6862 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
6863 \addtocounter{LWR@thisautoid}{1}%
6864 \booltrue{LWR@freezethisautoid}%
```

```
6865 \begingroup%
```

Settings while inside the environment:

```
6866 \LWR@origraggedright%
```

Open an HTML figure tag:

```
6867 \LWR@htmltag{figure id="\LWR@origmbox{autoid-\arabic{LWR@thisautoid}}" class="#1"}%
6868 \ifbool{FormatWP}{%
6869   \LWR@orignewline%
6870   \LWR@BlockClassWP}{\wp#1}%
6871 }{}%
```

```
6872 \renewcommand*{\@capttype}{#1}%
6873 \caption@settype{#1}%
6874 \LWR@startpars%
6875 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6876
6877   === begin #1 ===
6878
6879 }{}%
6880 }
```

\@float Support packages which create floats directly.
 \@dblfloat

```
6881 \let\@float\LWR@floatbegin
6882 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
6883 \newcommand*{\LWR@floatend}{%
6884 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
6885
6886 === end ===
6887
6888 }{}%
6889 \LWR@stoppars%
```

Close an HTML figure tag:

```
6890 \ifbool{FormatWP}{\endLWR@BlockClassWP}{}%
6891 \LWR@htmlElementend{figure}%
6892 \endgroup%
6893 \boolfalse{LWR@freezethisautoid}%
6894 \LWR@startpars%
6895 \ifbool{FormatWP}{\newline}{}%
6896 }
```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```
6897 \let\end@float\LWR@floatend
6898 \let\end@dblfloat\LWR@floatend
```

Ctr `LWR@thisautoid` A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

```
6899 \newcounter{LWR@thisautoid}
```

Ctr `LWR@thisautoidWP` A sequential counter for all word processor conversion `<div>`s. This is used to convince LibreOffice to form a frame around this element.

```
6900 \newcounter{LWR@thisautoidWP}
```

Bool `LWR@freezethisautoid` Prevents multiple increments of `\LWR@thisautoid` inside a float.

```
6901 \newbool{LWR@freezethisautoid}
6902 \boolfalse{LWR@freezethisautoid}
```

`\LWR@newautoidanchor` Adds a new `<autoid>` anchor.

```
6903 \newcommand*{\LWR@newautoidanchor}{%
6904 \ifbool{LWR@freezethisautoid}{}%
6905     \addtocounter{LWR@thisautoid}{1}%
6906     \LWR@htmltag{a id="\LWR@origmbbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
```

```
6907 }%
6908 }
```

`\@capttype` Remembers which float type is in use.

```
6909 \newcommand*\@capttype{}
```

65.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```
6910 \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.

```
6911 \AtBeginDocument{\renewcommand*\@makecaption}[2]{%
6912   \LWR@traceinfo{\@makecaption}%
6913   #1\CaptionSeparator#2%
6914   \LWR@traceinfo{\@makecaption: done}%
6915 }%
6916 }
```

65.1.2 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`, `.lot` file is read.

```
6917 \newcounter{LWR@nextautoid}
6918 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat` $\{\langle autopage \rangle\} \{\langle autoid \rangle\}$

This is written to the .lof, .lot file just before each float's usual entry. The autopage and autoid are remembered for \l@figure to use when creating the HTML links.

```
6919 \newcommand*\LWRsetnextfloat}[2]{%
6920 \setcounter{LWR@nextautopage}{#1}%
6921 \setcounter{LWR@nextautoid}{#2}%
6922 }
```

Ctr **LWR@latestautopage** Updated each time a new HTML file is begun. \LWRsetnextfloat is written with this and the autoid by the modified \addcontentsline just before each float's entry.

```
6923 \newcounter{LWR@latestautopage}
6924 \setcounter{LWR@latestautopage}{1}
```

Env **LWR@figcaption** Encapsulates a caption inside <figcaption>, and if FormatWP then also a <div> with an italic style.

```
6925 \newenvironment*LWR@figcaption
6926 {%
6927 \LWR@traceinfo{LWR@figcaption env start}%
6928 \LWR@htmlblocktag{figcaption}%
6929 \ifbool{FormatWP}{%
6930 \begin{BlockClass}[font-style:italic]{italic}
6931 \LWR@origvspace*\baselineskip
6932 }{}%
6933 \LWR@traceinfo{LWR@figcaption env start: done}%
6934 }
6935 {%
6936 \LWR@traceinfo{LWR@figcaption env end}%
6937 \ifbool{FormatWP}{\end{BlockClass}}{}%
6938 \LWR@htmlblocktag{/figcaption}%
6939 \LWR@traceinfo{LWR@figcaption env end: done}%
6940 }
```

After packages have loaded, remember the print-mode version of the following:

```
6941 \AtBeginDocument{
6942 \LetLtxMacro\LWR@origcaption@begin\caption@begin
6943 \LetLtxMacro\LWR@origcaption@end\caption@end
6944 }
```

\LWR@caption@begin Low-level patches to create HTML tags for captions.

```
6945 \newcommand*\LWR@caption@begin}[1]
6946 {
6947 \LWR@traceinfo{LWR@caption@begin}%
```

Keep par and minipage changes local:

```
6948 \begingroup%
```

The caption code was not allowing the closing par tag:

```
6949 \@setpar{\LWR@closeparagraph\@par}%
```

No need for a minipage or \parbox inside the caption:

```
6950 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
6951 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
6952 \LWR@figcaption%
6953 \LWR@traceinfo{\LWR@caption@begin: about to LWR@origcaption@begin}%
6954 \LWR@origcaption@begin{#1}%
6955 \LWR@traceinfo{\LWR@caption@begin: done}%
6956 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
6957 \newcommand{\LWR@caption@end}
6958 {%
6959 \LWR@traceinfo{\LWR@caption@end}%
6960 \LWR@origcaption@end%
```

Closing tag:

```
6961 \endLWR@figcaption%
6962 \endgroup%
6963 % \leavevmode% avoid bad space factor (0) error
6964 \LWR@traceinfo{\LWR@caption@end: done}%
6965 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.
`\caption@end`

```
6966 \AtBeginDocument{
6967 \let\caption@begin\LWR@caption@begin
6968 \let\caption@end\LWR@caption@end
6969 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

6970 \let\LWR@origcaptionlistentry\captionlistentry
6971
6972 \renewcommand*{\captionlistentry}{%
6973 \LWR@ensuredoingapar%
6974 \LWR@origcaptionlistentry%
6975 }
6976
6977 \def\LWR@LTcaptionlistentry{%
6978 \LWR@ensuredoingapar%
6979 \LWR@htmltag{a id="\LWR@origmbox{autoid-\arabic{\LWR@thisautoid}}"}\LWR@htmltag{/a}%
6980 \bgroup
6981 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
6982 {\egroup\LWR@LTcaptionlistentry}}%
6983 \def\LWR@LTcaptionlistentry#1{%
6984 \caption@listentry\@firstoftwo[\LTcctype]{#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.

```

6985 \let\LWR@origaddcontentsline\addcontentsline
6986
6987 \renewcommand*{\addcontentsline}[3]{%
6988 \ifstrequal{#1}{toc}{}% not TOC
6989 \LWR@newautoidanchor%
6990 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{}%
6991 \addtocontents{\@nameuse{ext@#2}}{%
6992 \protect\LWRsetnextfloat%
6993 {\arabic{\LWR@latestautopage}}%
6994 {\arabic{\LWR@thisautoid}}%
6995 }%
6996 }% not TOC
6997 \LWR@origaddcontentsline{#1}{#2}{#3}%
6998 }

```

Pkg `capt-of` Either package provides `\captionof`, which is later patched at the beginning of the document.

Pkg `caption`

`\captionof` Patched to handle paragraph tags.

```

6999 \AtBeginDocument{
7000 \let\LWR@origcaptionof\captionof
7001
7002 \renewcommand*{\captionof}{%
7003 \LWR@stoppars
7004 \LWR@origcaptionof%
7005 }
7006 }

```

```
7007 \end{warpHTML}
```

66 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: 7008 \begin{warpHTML}

66.1 Reading and printing the TOC

```
\LWR@myshorttoc {<toc/lof/lot/sidetoc>}
```

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the @ character into a normal letter to allow formatting commands in the section names.

Unlike in regular \TeX , the file is not reset after being read, since the sideroc may be referred to again in each HTML page.

```
7009 \newcommand*{\LWR@myshorttoc}[1]{%
7010 \LWR@traceinfo{\LWR@myshorttoc: #1}%
7011 \LWR@ensuredoingapar%
```

Only if the file exists:

```
7012 \IfFileExists{\jobname.#1}{%
7013 \LWR@traceinfo{\LWR@myshorttoc: loading}%
```



Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.



disabled

For pdf_latex, 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.

```

7014 \begingroup%
7015 % \ifxetexorluatex%
7016 % \else
7017 % \inputencoding{latin1}% currently disabled
7018 % \fi
7019 \makeatletter%

```

Read in the TOC file:

```

7020 \@input{\jobname.#1}%
7021 % \makeatother
7022 \endgroup%
7023 }%
7024 {}%
7025 \LWR@traceinfo{LWR@myshorttoc: done}%
7026 }

```

`\LWR@subtableofcontents` $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```

7027 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%

```

Closes previous levels:

```

7028 \@ifundefined{chapter}
7029 {\LWR@closeprevious{\LWR@depthsection}}
7030 {\LWR@closeprevious{\LWR@depthchapter}}

```

Prints any pending footnotes so that they appear above the potentially large TOC:

```

7031 \LWR@printpendingfootnotes

```

Place the list into its own chapter (if defined) or section:

```

7032 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}

```

Create a new HTML nav containing the TOC/LOF/LOT:

```

7033 \LWR@htmlclass{nav}{#1}

```

Create the actual list:

```

7034 \LWR@myshorttoc{#1}

```

Close the nav:

```

7035 \LWR@html elementclassend{nav}{#1}
7036 }

```

`\@starttoc` `{\langle ext\rangle}`

Patch `\@starttoc` to encapsulate the toc inside HTML tags:

```

7037 \let\LWR@orig@starttoc\@starttoc
7038
7039 \renewcommand{\@starttoc}[1]{
7040 \LWR@html elementclass{nav}{#1}
7041 \LWR@orig@starttoc{#1}
7042 \LWR@html elementclassend{nav}{#1}
7043 }

```

Bool `LWR@copiedsidetoc` Used to only copy the toc file to the sidetoc a single time.

(listings and perhaps other packages would re-use `\tableofcontents` for their own purposes, causing the sidetoc to be copied more than once, and thus end up empty.)

```

7044 \newbool{LWR@copiedsidetoc}
7045 \boolfalse{LWR@copiedsidetoc}

```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```

7046 \AtBeginDocument{
7047 \let\LWR@origtableofcontents\tableofcontents
7048
7049 \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:

```

7050 \ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{
7051
7052 === table of contents ===
7053
7054 }
7055 {

```

Copy the `.toc` file to `.sidetoc` for printing the sideroc. The original `.toc` file is renewed when `\tableofcontents` is finished.

```

7056 \ifbool{LWR@copiedsidetoc}{\{%
7057 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%

```

```

7058      \booltrue{LWR@copiedsidetoc}%
7059    }%
7060    \LWR@printpendingfootnotes
7061    \LWR@origtableofcontents
7062 }
7063 }% \tableofcontents
7064 }% AtBeginDocument

```

`\listoffigures`

```

7065 \let\LWR@origlistoffigures\listoffigures
7066
7067 \renewcommand*{\listoffigures}{
7068 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
7069
7070 === list of figures ===
7071
7072 }
7073 {
7074   \LWR@printpendingfootnotes
7075   \LWR@origlistoffigures
7076 }
7077 }

```

`\listoftables`

```

7078 \let\LWR@origlistoftables\listoftables
7079
7080 \renewcommand*{\listoftables}{
7081 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
7082
7083 === list of tables ===
7084
7085 }
7086 {
7087   \LWR@printpendingfootnotes
7088   \LWR@origlistoftables
7089 }
7090 }

```

66.2 High-level TOC commands

`\listof` $\{\langle type \rangle\} \{\langle title \rangle\}$

Emulate the `\listof` command from the float package (section 161). Used to create lists of custom float types. Also used to redefine the standard \LaTeX `\listoffigures` and `\listoftables` commands.

```

7091 \NewDocumentCommand{\listof}{m +m}{%
7092 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
7093 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
7094 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
7095     \jobname.\csuse{ext@#1}\relax
7096 }
```

66.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

css may be used to format the sideTOC:

CSS related to sideTOC:

nav.sidetoc: The entire sideroc.

div.sidetoctitle: The title.

div.sidetoccontents: The table of contents.

```

7097 \end{warpHTML}
```

for HTML & PRINT: 7098 `\begin{warpall}`

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard \LaTeX section level similar to `tocdepth`.

```

7099 \newcounter{SideTOCDepth}
7100 \setcounter{SideTOCDepth}{1}
```

`\sidetocname` Holds the default name for the sideroc.

```

7101 \newcommand{\sidetocname}{Contents}
```

```

7102 \end{warpall}
```

for HTML output: 7103 `\begin{warpHTML}`

`\LWR@sidetoc` Creates the actual side-TOC.

```
7104 \newcommand*{\LWR@sidetoc}{
7105 \LWR@forcenewpage
7106 \LWR@stoppars
7107 }
```

The entire siderOC is placed into a nav of class sidetoc.

```
7108 \LWR@htmlclass{nav}{sidetoc}
7109
7110 \setcounter{tocdepth}{\value{SideTOCDepth}}
7111 }
```

The title is placed into a <div> of class sidetoc title, and may contain paragraphs.

```
7112 \begin{BlockClass}{sidetoc title}
7113 \sidetocname
7114 \end{BlockClass}
```

The table of contents is placed into a <div> of class sidetoc contents.

```
7115 \begin{BlockClass}{sidetoc contents}
7116 \LinkHome
7117
7118 \LWR@myshorttoc{sidetoc}
7119 \end{BlockClass}
7120 \LWR@htmlclassend{nav}{sidetoc}
7121 }
```

66.4 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:

```
7122 \newcommand*{\LWR@numberline}[1]{%
7123 \LWR@sectionnumber{#1}\quad%
7124 }
7125
7126 \LetLtxMacro\numberline\LWR@numberline
```

`\hypertoc {<1: depth>} {<2: type>} {<3: name>} {<4: page>}`

Called by `\l@section`, etc. to create a hyperlink to a section.

The autopage label is always created just after the section opens.

#1 is depth

#2 is section, subsection, etc.

#3 the text of the caption

#4 page number

```
7127 \NewDocumentCommand{\hypertoc}{m m +m m}{%
7128 \LWR@traceinfo{hypertoc !#1!#2!#3!#4!}%
```

Respond to `tocdepth`:

```
7129 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}{%
7130 \LWR@startpars%
```

Create an HTML link to `filename#autosec-(page)`, with text of the caption, of the given HTML class.

```
7131 \LWR@subhyperrefclass{%
7132 \LWR@htmlrefsectionfilename{autopage-#4}\#\LWR@origmbox{autosec-#4}%
7133 }{#3}{toc#2}%
7134 \LWR@stoppars%
7135 }
7136 {}
7137 \LWR@traceinfo{hypertoc done}%
7138 }
```

Ctr `lofdepth` TOC depth for figures.

```
7139 \@ifclassloaded{memoir}{%{
7140 \newcounter{lofdepth}
7141 \setcounter{lofdepth}{1}
7142 }
```

Ctr `lotdepth` TOC depth for tables.

```
7143 \@ifclassloaded{memoir}{%{
7144 \newcounter{lotdepth}
7145 \setcounter{lotdepth}{1}
7146 }
```

`\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

```
7147 \newcommand{\hypertocfloat}[5]{%
7148 \LWR@startpars
```

If some float-creation package has not yet defined the float type's lofdepth counter, etc, define it here:

```
7149 \@ifundefined{c@#3depth}{%
7150 \newcounter{#3depth}%
7151 \setcounter{#3depth}{1}%
7152 }{}%
```

Respond to lofdepth, etc.:

```
7153 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
7154 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{}%
7155 \LWR@startpars%
```

Create an HTML link to filename#autoid-(float number), with text of the caption, of the given HTML class.

```
7156 \LWR@subhyperrefclass{%
7157 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
7158 \#\LWR@origmbox{autoid-\arabic{LWR@nextautoid}}}%
7159 {#4}{toc#2}%
7160 \LWR@stoppars%
7161 }{}%
7162 }
```

Automatically called by `\contentsline`:

`\l@part` $\{\langle name \rangle\} \{\langle page \rangle\}$

Uses `\DeclareDocumentCommand` in case the class does not happen to have a `\part`.

```
7163 \DeclareDocumentCommand{\l@part}{m m}{\hypertoc{-1}{part}{#1}{#2}}
```

`\l@chapter` $\{\langle name \rangle\} \{\langle page \rangle\}$

Uses `\DeclareDocumentCommand` in case the class does not happen to have a `\chapter`.

```
7164 \DeclareDocumentCommand{\l@chapter}{m m}
7165   {\hypertoc{0}{chapter}{#1}{#2}}
```

`\l@section` $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7166 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
```

`\l@subsection` $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7167 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
```

`\l@subsubsection` $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7168 \renewcommand{\l@subsubsection}[2]{\hypertoc{3}{subsubsection}{#1}{#2}}
```

`\l@paragraph` $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7169 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
```

`\l@subparagraph` $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7170 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
```

`\l@figure` $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7171 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
```

`\l@table` $\{\langle name \rangle\} \{\langle page \rangle\}$

```
7172 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}
```

```
7173 \end{warpHTML}
```

67 Index and glossary

See:

<http://tex.stackexchange.com/questions/187038/>

[how-to-mention-section-number-in-index-created-by-imakeidx](#)

Index links are tracked by the counter LWR@autoindex. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output: 7174 \begin{warpHTML}

```
7175 \newcounter{LWR@autoindex}
7176 \setcounter{LWR@autoindex}{0}
7177
7178 \newcounter{LWR@autoglossary}
7179 \setcounter{LWR@autoglossary}{0}
```

\printindex

```
7180 \let\LWR@origprintindex\printindex
7181
7182 \renewcommand*\printindex{
7183 {
7184 \LWR@startpars
7185 \LWR@origprintindex
7186 }
```

Env theindex

```
7187 \@ifundefined{chapter}
7188 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}
7189 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}
7190
7191 \renewenvironment*{theindex}{%
7192 \LWR@indexsection{\indexname}%
7193 \let\item\LWR@indexitem%
7194 \let\subitem\LWR@indexsubitem%
7195 \let\subsubitem\LWR@indexsubsubitem%
7196 }{}
```

\LWR@indexitem

```
7197 \newcommand*\LWR@indexitem){
7198
7199 \InlineClass{indexitem}{}
7200 }
```

\LWR@indexitem

```
7201 \newcommand*\LWR@indexsubitem){
```

```

7202
7203 \InlineClass{indexsubitem}{ }
7204 }

```

\LWR@indexitem

```

7205 \newcommand{\LWR@indexsubsubitem}{
7206
7207 \InlineClass{indexsubsubitem}{ }
7208 }

```

\@wrindex {<term>} Redefined to write the LWR@autoindex counter instead of page

```

7209 \def\LWR@wrindex#1{%
7210 \addtocounter{LWR@autoindex}{1}%
7211 \LWR@newlabel{LWRindex-\arabic{LWR@autoindex}}{%
7212 \protected@write\@indexfile{ }%
7213 {\string\indexentry{#1}{\arabic{LWR@autoindex}}}%
7214 \endgroup
7215 \@esphack}
7216
7217 \let\@wrindex\LWR@wrindex

```

\@wrglossary {<term>} Redefined to write the LWR@latestautopage counter instead of page

```

7218 \def\@wrglossary#1{%
7219 \addtocounter{LWR@autoglossary}{1}%
7220 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
7221 \protected@write\@glossaryfile{ }%
7222 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
7223 \endgroup
7224 \@esphack}

```

\hyperindexref {<autosecnumber>}

\hyperindexref{web address} is inserted into *.ind by the xindy style file lwarp.xdy

```

7225 \newcommand*\hyperindexref[1]{\nameref{LWRindex-#1}}

7226 \end{warpHTML}

```

for PRINT output: A null command for print mode, in case hyperref was not used:

```

7227 \begin{warpprint}
7228 \newcommand{\hyperindexref}[1]{#1}
7229 \end{warpprint}

```

for HTML & PRINT: For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
7230 \begin{warpall}
7231 \providecommand{\glo@name}{}
7232 \end{warpall}
```

68 Bibliography presentation

for HTML output: 7233 `\begin{warpHTML}`

`\bibliography` `{\filenames)}`

Modified to use the base jobname instead of the `_html` jobname.

```
7234 \def\bibliography#1{%
7235     \if@filesw
7236     \immediate\write\@auxout{\string\bibdata{#1}}%
7237     \fi
7238 %     \@input@{\jobname.bbl}% original
7239     \@input@{\BaseJobname.bbl}% lwarp
7240 }
```

`\@biblabel` `{\text-refnumber)}`

Modified to use the base jobname instead of the `_html` jobname.

```
7241 \renewcommand{\@biblabel}[1]{[#1]\quad}
```

Env `thebibliography` To emphasize document titles in the bibliography, the following redefines `\em` inside `thebibliography` to gather everything until the next closing brace, then display these tokens with `\textit`.

Adapted from `embracedef.sty`, which is by TAKAYUKI YATO:

<https://gist.github.com/zr-tex8r/b72555e3e7ad2f0a37f1>

```
7242 \AtBeginDocument{
7243 \AtBeginEnvironment{thebibliography}{
7244 \providecommand*\LWR@newem}[1]{\textit{#1}}
7245
7246 \renewrobustcmd{\em}{%
7247     \begingroup
7248     \gdef\LWR@em@after{\LWR@em@finish\LWR@newem}%
7249     \afterassignment\LWR@em@after
7250     \toks@\bgroup
```

```

7251 }
7252
7253 \def\LWR@em@finish#1{%
7254     \xdef\LWR@em@after{\noexpand#1{\the\toks@}}%
7255     \endgroup
7256     \LWR@em@after\egroup
7257 }
7258 }% \AtBeginEnvironment{thebibliography}
7259 }% \AtBeginDocument

7260 \end{warpHTML}

```

69 Restoring original formatting

for HTML output: 7261 \begin{warpHTML}

`\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`.

```

7262 \newcommand*{\LWR@restoreorigformatting}{%
7263 \LWR@traceinfo{\LWR@restoreorigformatting}%
7264 \linespread{1}%

7265 \LetLtxMacro\caption@begin\LWR@origcaption@begin%
7266 \LetLtxMacro\caption@end\LWR@origcaption@end%
7267 \let\par\LWR@origpar%

7268 \LetLtxMacro\ref\LWR@origref%{ } syntax highlighting

7269 \let\normalsize\LWR@orignormalsize%
7270 \let\small\LWR@origsmall%
7271 \let\footnotesize\LWR@origfootnotesize%
7272 \let\scriptsize\LWR@origscriptsize%
7273 \let\tiny\LWR@origtiny%
7274 \let\large\LWR@origlarge%
7275 \let\Large\LWR@origLarge%
7276 \let\LARGE\LWR@origLARGE%
7277 \let\huge\LWR@orighuge%
7278 \let\Huge\LWR@origHuge%

```

```

7279 \RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
7280 \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
7281 \renewcommand{\BlockClassSingle}[2]{##2}%
7282 \LetLtxMacro{\hspace}{\LWR@orighspace}%

7283 \LetLtxMacro\hfill\LWR@orighfill%
7284 \LetLtxMacro\hfil\LWR@orighfil%
7285 \LetLtxMacro\rule\LWR@origrule%
7286 \LetLtxMacro\hrulefill\LWR@orighrulefill%
7287 \LetLtxMacro\dotfill\LWR@origdotfill%
7288 \let\vspace\LWR@origvspace%
7289 \let\hss\LWR@orighss%
7290 \let\llap\LWR@origllap%
7291 \let\rlap\LWR@origrlap%
7292 \let\hfilneg\LWR@origfilneg%

7293 \let\,\LWR@origcomma% disable HTML short unbreakable space
7294 \let\textellipsis\LWR@origtextellipsis%
7295 \let\textless\LWR@origtextless%
7296 \let\textgreater\LWR@origtextgreater%
7297 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
7298 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
7299 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
7300 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
7301 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
7302 \LetLtxMacro{\textit}{\LWR@origtextit}%
7303 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
7304 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
7305 \LetLtxMacro{\textup}{\LWR@origtextup}%
7306 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
7307 \LetLtxMacro{\emph}{\LWR@origemph}%
7308 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
7309 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
7310 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
7311 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
7312 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
7313 \LetLtxMacro{\upshape}{\LWR@origupshape}%
7314 \LetLtxMacro{\slshape}{\LWR@origslshape}%
7315 \LetLtxMacro{\scshape}{\LWR@origscshape}%
7316 \LetLtxMacro{\itshape}{\LWR@origitshape}%
7317 \LetLtxMacro{\em}{\LWR@origem}%
7318 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
7319 \let\sp\LWR@origsp%
7320 \let\sb\LWR@origsb%
7321 \LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
7322 \LetLtxMacro@textsuperscript\LWR@orig@textsuperscript%
7323 \LetLtxMacro\textsubscript\LWR@origtextsubscript%
7324 \LetLtxMacro@textsubscript\LWR@orig@textsubscript%
7325 \LetLtxMacro\underline\LWR@origunderline%

```

```

7326 \let~\LWR@origtilde%
7327 \let\enskip\LWR@origenskip%
7328 \let\quad\LWR@origquad%
7329 \let\qquad\LWR@origqquad%
7330 \LetLtxMacro\tabular\LWR@origtabular%
7331 \LetLtxMacro\endtabular\LWR@origendtabular%
7332 \LetLtxMacro\noalign\LWR@orignoalign%
7333 \LetLtxMacro\hline\LWR@orighline%
7334 \LetLtxMacro\toprule\LWR@origtoprule%
7335 \LetLtxMacro\midrule\LWR@origmidrule%
7336 \LetLtxMacro\cmidrule\LWR@origcmidrule%
7337 \LetLtxMacro\bottomrule\LWR@origbottomrule%
7338 \LetLtxMacro\addlinespace\LWR@origaddlinespace%
7339 \LetLtxMacro\morecmidrules\LWR@origmorecmidrules%
7340 \LetLtxMacro\specialrule\LWR@origspecialrule%
7341 \let\newline\LWR@orignewline%
7342 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
7343 \LetLtxMacro\includegraphics\LWR@origincludegraphics%
7344 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
7345 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
7346 \let\reflectbox\LWR@origreflectbox%
7347 \LetLtxMacro\resizebox\LWR@origresizebox%
7348 \let\framebox\LWR@origframebox%

7349 \LetLtxMacro\mbox\LWR@origmbox%

7350 \let\makebox\LWR@origmakebox%
7351 \let\fbbox\LWRprint@fbbox%
7352 \let\fbboxBlock\LWRprint@fbbox%
7353 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
7354 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
7355 \LetLtxMacro{\minipage}{\LWR@origminipage}%
7356 \let\endminipage\LWR@origendminipage%
7357 \LetLtxMacro{\parbox}{\LWR@origparbox}%
7358 \let\TeX\LWR@origTeX%
7359 \let\LaTeX\LWR@origLaTeX%
7360 \let\LaTeXe\LWR@origLaTeXe%
7361 \renewcommand*{\Xe}{X\textsubscript{E}}%

7362 \LetLtxMacro\@ensuredmath\LWR@origensuredmath%

7363 \csletcs{equation*}{LWR@origequationstar}%
7364 \csletcs{endequation*}{LWR@origendequationstar}%
7365 %
7366 \LWR@restoreorigaccents%
7367 \LWR@restoreoriglists%
7368 %
7369 \LWR@FBcancel%
```

7370 }

7371 \end{warpHTML}

70 Math

70.1 Limitations

70.1.1 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.

70.1.2 SVG option


SVG math option For SVG math, math is rendered as usual by \LaTeX into the initial PDF file using the current font¹⁵, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML

¹⁵See section 321 regarding fonts and fractions.

`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.

 **SVG math in \TeX boxes** SVG math does not work inside \TeX boxes, since a `\newpage` is required before and after each image.

70.1.3 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
2. As detokenized printed \TeX commands placed directly into the HTML output for interpretation by the MATHJAX display scripts. An additional script is used to pre-set the equation number format and value according to the current \TeX values, and the MATHJAX cross-referencing system is ignored in favor of the \TeX internal system, seamlessly integrating with the rest of the \TeX code.

MATHJAX limitations Limitations when using MATHJAX include:

Prog MathJax

chapter numbers

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

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

subequations

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

footnotes in math

- 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.

⚠ `siunitx` inside an equation

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

Also see section 8.5.5.

tabbing

- A `tabbing` environment is emulated using an HTML `<pre>`. While MATHJAX is enabled inside `tabbing`, the browser may not correctly render the horizontal alignment of the math and text following after on the same line.

⚠ other macros and packages

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

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}
```

70.2 Inline and display math

for HTML output: 7372 `\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.

```

7373 \let\LWR@origtextdollar\$
7374
7375 \renewcommand*{\$}{%
7376 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
7377 {\LWR@origtextdollar}%
7378 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
7379 }

```

Ctr LWR@externalfilecnt Counter for the external files which are generated and then referenced from the HTML:

```

7380 \newcounter{LWR@externalfilecnt}

7381 \LetLtxMacro\LWR@origdollar$
7382 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

7383 \LetLtxMacro\LWR@origopenparen\ (
7384 \LetLtxMacro\LWR@origcloseparen\)
7385 \LetLtxMacro\LWR@origopenbracket\[
7386 \LetLtxMacro\LWR@origclosebracket\]

```

\$ Redefine the dollar sign to place math inside a lateximage, or use MATHJAX:
 \$\$

```

7387 \begingroup
7388 \catcode'\$=\active%
7389 \protected\gdef$\{\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

\LWR@doubledollar Redefine the double dollar sign to place math inside a lateximage, or use MATHJAX:

```

7390 \protected\gdef\LWR@doubledollar$#1$${

```

If MATHJAX or formatting for a word processor, print the \TeX expression:

```

7391 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%

```

For MATHJAX, print the math between \[and \]:

```

7392 {
7393
7394     \textbackslash[
7395     \LWR@HTMLsanitize{#1}%
7396     \textbackslash]
7397
7398 }% mathjax

```

For SVG, print the math inside a lateximage, with an <alt> tag of the \LaTeX code:

```

7399 {% not mathjax
7400
7401     \begin{lateximage}%
7402     [\textbackslash{[] \LWR@HTMLsanitize{#1} \textbackslash{[]}]%
7403     \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
7404     \end{lateximage}%
7405
7406 }% not mathjax
7407 }%
```

\backslash LWR@singledollar Redefine the single dollar sign to place math inside a lateximage, or use MATHJAX:

```

7408 \newlength{\LWR@singledollarwidth}
7409 \newlength{\LWR@singledollarheight}
7410 \newlength{\LWR@singledollardepth}
7411
7412 \protected\gdef\LWR@subsingledollar#1{%
7413 \begingroup%
7414 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
```

For MATHJAX, print the math between \backslash (and \backslash):

```

7415 {%
7416     {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}%
7417 }% mathjax
```

For SVG, print the math inside a lateximage, with an <alt> tag of the \LaTeX code, and a css style to control the baseline adjustment.

```

7418 {% not mathjax
```

Measure the depth, width, and height of the math image:

```

7419     \LWR@restoreorigformatting%
7420     \LWR@orignormalsize%
7421     \global\setlength{\LWR@singledollardepth}{\depthof{%
7422         \LWR@origdollar#1\LWR@origdollar%
7423     }*\real{.8}}%
7424     \global\setlength{\LWR@singledollarwidth}{\widthof{%
7425         \LWR@origdollar#1\LWR@origdollar%
7426     }*\real{.8}}%
7427     \global\setlength{\LWR@singledollarheight}{\totalheightof{%
7428         \LWR@origdollar#1\LWR@origdollar%
7429     }*\real{.8}}%
7430
7430     \LWR@origscriptsize%
```

Set a style for the the height or width. The em unit is used so that the math scales according to the user's selected font size. Start with the larger of width or height:

```

7431 %      \ifdimgreater{\LWR@singledollarwidth}{\LWR@singledollarheight}{%
7432 %          \def\LWR@singledollarstyle{%
7433 %              width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
7434 %          }%
7435 %      }{%
7436          \def\LWR@singledollarstyle{%
7437              height:\LWR@convertto{em}{\the\LWR@singledollarheight }em%
7438          }%
7439 %      }%

```

If narrow width, use the height. Single-letter variables look best if they all are scaled according to height.

```

7440 %      \ifdimless{\LWR@singledollarwidth}{.8em}%
7441 %      {%
7442 %          \def\LWR@singledollarstyle{%
7443 %              height:\LWR@convertto{em}{\the\LWR@singledollarheight }em%
7444 %          }%
7445 %      }%
7446 %      {}%

```

If extremely thin, use the width:

```

7447      \ifdimless{\LWR@singledollarheight}{.3em}%
7448      {%
7449          \def\LWR@singledollarstyle{%
7450              width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
7451          }%
7452      }%
7453      {}%

```

If there is significant text depth, add the depth to the style.

```

7454      \ifdimgreater{\LWR@singledollardepth}{0.01em}{%
7455          \def\LWR@singledollardepthstyle{%
7456              \ ; \LWR@origmbox{vertical-align:-\LWR@convertto{em}{\the\LWR@singledollardepth} em%
7457          }%
7458      }{%
7459          \def\LWR@singledollardepthstyle{}%
7460      }%

```

Create the lateximage using the alternate tag and the computed size and depth.

```

7461      \begin{lateximage}%
7462          [\textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)]%
7463          [\LWR@singledollarstyle \LWR@singledollardepthstyle]%

```

```

7464 \LWR@origdollar#1\LWR@origdollar%
7465 \end{lateximage}%
7466 }%
7467 \endgroup%
7468 }
7469
7470 \protected\gdef\LWR@singledollar#1${%
7471 \LWR@subsingledollar{#1}%
7472 }

```

\(Redefine to the above dollar macros.
 \)

```

7473 \protected\gdef\(#1\){$#1$}
7474 \protected\gdef\[#1\]{$$#1$$}
7475
7476 \endgroup

```

\@ensuredmath $\langle expression \rangle$ Not yet adapted to lwarp.

```

7477 \LetLtxMacro\LWR@origensuredmath\@ensuredmath
7478
7479 \renewcommand{\@ensuredmath}[1]{%
7480 \ifmmode%
7481 \LWR@origensuredmath{#1}%
7482 \else%
7483 \LWR@subsingledollar{\relax#1}%
7484 \fi%
7485 }

```

Remove the old math and displaymath environments:

```

7486 \let\math\relax
7487 \let\endmath\relax
7488 \let\displaymath\relax
7489 \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.

```

7490 \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.

```

7491 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}

```

70.3 MATHJAX support

Ctrl LWR@nextequation Used to add one to compute the next equation number.

```
7492 \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 .

```
7493 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
7494 \ifcsdef{thechapter}{
7495 \InlineClass{hidden}{
7496 \textbackslash(
7497 \textbackslash)}seteqsection \{\thechapter\}
7498 \textbackslash)
7499 }
7500 }
7501 \}{}% not using chapters
```

MATHJAX doesn't allow setting the equation number to 1:

```
7502 \ifthenelse{\cnttest{\value{equation}}>0}
7503 {
```

Tell MATHJAX that the next set of equations begins with the current \TeX equation number, plus one.

```
7504 \setcounter{LWR@nextequation}{\value{equation}}
7505 \addtocounter{LWR@nextequation}{1}
```

Place the MATHJAX command inside “\(" and “\)” characters, to be printed to HTML, not interpreted by \TeX .

```
7506 \InlineClass{hidden}{
7507 \textbackslash(
7508 \textbackslash)}seteqnumber \{\arabic{LWR@nextequation}\}
7509 \textbackslash)
7510 }
7511 \}{}% not eq > 0
7512 }
```

`\LWR@hidelatexequation` $\{\langle environment \rangle\} \{\langle contents \rangle\}$

Creates the \LaTeX version of the equation inside an HTML comment.

```
7513 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
7514 \LWR@stoppars
7515 \LWR@htmlopencomment
7516
```

Start the \LaTeX math environment inside the HTML comment:

```
7517 \begingroup
7518 \csuse{\LWR@orig#1}
```

While in the math environment, restore various commands to their \LaTeX meanings.

```
7519 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 70.5.1.

Print the contents of the equation:

```
7520 #2
```

End the \LaTeX math environment inside the HTML comment:

```
7521 \csuse{\LWR@origend#1}
7522 \endgroup
7523
```

Close the HTML comment and resume HTML paragraph handling:

```
7524 \LWR@htmlclosecomment
7525 \LWR@startpars
7526 }
```

`\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 .

```
7527 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the `MATHJAX` environment inside printed “\(" and “\)” characters.

```
7528 \LWR@origtilde\LWR@orignewline
7529 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
7530 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MATHJAX environment:

```
7531 \textbackslash{}end\{#1\}
7532 \LWR@orignewline
7533 }
```

70.4 Equation environment

Remember existing equation environment:

```
7534 \let\LWR@origequation\equation
7535 \let\LWR@origendequation\endequation
```

Remove existing equation environment:

```
7536 \let\equation\relax
7537 \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.

```
7538 \NewEnviron{equation}{%
7539
```

If `mathjax` or `FormatWP`, print the \LaTeX expression:

```
7540 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }{%
```

MATHJAX output:

7541 {

Print commands to synchronize MATHJAX's equation number and format to the current \LaTeX chapter/section and equation number:

7542 \LWR@syncmathjax

Print the \LaTeX math inside an HTML comment:

7543 \LWR@hidelatexequation{equation}{\BODY}

7544 }

SVG output: Create the lateximage along with an HTML <alt> tag having an equation number, the \LaTeX equation environment commands, and the contents of the environment's \BODY.

7545 {% not mathjax

Begin the lateximage with an <alt> tag containing the math source:

7546 \begin{lateximage}[({\LWR@equationtag} \textbackslash{begin}\{equation\})] %

7547 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}} %

7548 \textbackslash{end}\{equation\})]% alt tag

Create the actual \LaTeX -formatted equation inside the lateximage using the contents of the environment.

7549 \LWR@origequation

7550 \BODY% contents collected by NewEnviron

7551 \LWR@origendequation

7552 \end{lateximage}%

7553 }% not mathjax

7554

After the environment, if MATHJAX, print the math to the HTML output for MATHJAX processing:

7555 }[%

7556 \ifbool{mathjax} or (bool{FormatWP} and bool{WPMarkMath})]%

7557 {%

7558 \LWR@addmathjax{equation}{\BODY}%

7559 }%]

7560

7561]

Env equation*

```

7562 \csletcs{LWR@origequationstar}{equation*}
7563 \csletcs{LWR@origendequationstar}{endequation*}
7564 \renewenvironment*{equation*}
7565 {\displaymath}
7566 {\enddisplaymath}

```

70.5 AMS Math environments

70.5.1 Support macros

Bool LWR@amsmultline True if processing a multiline environment.

To compensate for multiline-specific code, LWR@amsmultline is used to add extra horizontal space in `\LWR@htmlmathlabel` if is used in an amsmath environment which is not a multiline environment and not an equation.

```

7567 \newbool{LWR@amsmultline}
7568 \boolfalse{LWR@amsmultline}

```

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

lwarp points `\ltx@label` here. This is used by `\label` when inside a \TeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the \TeX original, modified by lwarp, then by amsmath, then by cleveref.

```

7569 \newcommand*{\LWR@htmlmathlabel}[1]{%
7570 \LWR@traceinfo{LWR@htmlmathlabelb #1}%

```

If mathjax or FormatWP, print the \TeX expression:

```

7571 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7572 {%

```

The combined \TeX & HTML label is printed in a `\text` field:

```

7573 \text{

```

Shift the label over to the right side of the environment to avoid over-printing the math:

```

7574 \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.)

```

7575 \LWR@htmlclosecomment%
7576 \LWR@origltx@label{#1}%
7577 \LWR@htmlopencomment%
7578 }% text
7579 }% mathjax
7580 {%
7581 \LWR@origltx@label{#1}%
7582 }%
7583 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```

7584 \newcommand*{\LWR@beginhideamsmath}{
7585 \LWR@stoppars
7586 \LWR@origtilde\LWR@orignewline
7587 \LWR@htmlopencomment
7588
7589 \begingroup
7590 \LWR@restoreorigformatting
7591 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```

7592 \newcommand*{\LWR@endhideamsmath}{
7593 \endgroup
7594
7595 \LWR@htmlclosecomment
7596 \LWR@orignewline
7597 \LWR@startpars
7598 }
```

70.5.2 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 \LaTeX 's equation numbers, typeset with \LaTeX inside an HTML comment, then printed to HTML output for MATHJAX to process.

Env `eqnarray` This environmnet is not an \mathcal{AMS} environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into `\BODY`.

```
7599 \let\LWR@origeqnarray\eqnarray
7600 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
7601 \newbool{LWR@numbereqnarray}
7602 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
7603 \newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the \LaTeX expression:

```
7604 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7605 {%
```

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:

```
7606 \LWR@syncmathjax
7607 \boolfalse{LWR@amsmultline}
7608 \ifbool{LWR@numbereqnarray}
7609 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
7610 \LWR@beginhideamsmath
7611 \LWR@origeqnarray
7612 \BODY
7613 \LWR@origendeqnarray
7614 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MATHJAX to interpret:

```
7615 \LWR@addmathjax{eqnarray}{\BODY}
7616 }%
7617 {% not LWR@numbereqnarray
```

If not numbering equations, just create the contents for MATHJAX:

```

7618 \LWR@addmathjax{eqnarray*}{\BODY}
7619 }% LWR@numbreqnarray
7620 }% mathjax
7621 {% not mathjax
7622 \ifbool{LWR@numbreqnarray}
7623 {

```

For numbered svg equations, first create a lateximage with an alt attribute containing sanitized copy of the source code:

```

7624 \begin{lateximage}[(\LWR@startingequationtag--\LWR@equationtag)
7625 \LWR@addmathjax{eqnarray}{\BODY}]

```

Then create the image contents using an actual eqnarray:

```

7626 \LWR@origeqnarray
7627 \BODY
7628 \LWR@origendeqnarray
7629 \end{lateximage}
7630 }%
7631 {% not LWR@numbreqnarray

```

If not numbered, do the same, but an extra \nonumber seems to be required:

```

7632 \begin{lateximage}[(\LWR@addmathjax{eqnarray*}{\BODY})]
7633 \LWR@origeqnarray
7634 \BODY
7635 \nonumber
7636 \LWR@origendeqnarray
7637 \end{lateximage}
7638 }% LWR@numbreqnarray
7639 }% not mathjax

```

Default to number equations in the future:

```

7640 \booltrue{LWR@numbreqnarray}
7641 }

```

eqnarray itself is made with a blank line before and after to force it to be on its own line:

```

7642 \RenewEnviron{eqnarray}
7643 {%
7644
7645 \LWR@eqnarrayfactor
7646
7647 }

```

The starred version is patched to turn off the numbering:

```
7648 \csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}
```

The following $\mathcal{A}\mathcal{M}\mathcal{S}$ environments are more easily patched in-place:

Env `multline`

```
7649 \BeforeBeginEnvironment{multline}{
7650
7651 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7652 {
7653   \LWR@syncmathjax
7654   \booltrue{LWR@amsmultline}
7655   \LWR@beginhideamsmath
7656 }
7657 {
7658   \begin{lateximage}[\LWR@amsmathbodynumbered{multline}]
7659 }
7660 }
7661
7662 \AfterEndEnvironment{multline}{
7663
7664 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7665 {
7666   \LWR@endhideamsmath
7667   \boolfalse{LWR@amsmultline}
7668   \LWR@addmathjax{multline}{\the\@envbody}
7669 }
7670 {\end{lateximage}}
7671
7672 }
```

Env `multline*`

```
7673 \BeforeBeginEnvironment{multline*}{
7674
7675 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7676 {
7677   \LWR@syncmathjax
7678   \booltrue{LWR@amsmultline}
7679   \LWR@beginhideamsmath
7680 }
7681 {
7682   \begin{lateximage}[\LWR@amsmathbody{multline*}]
7683 }
7684 }
7685
```

```

7686 \AfterEndEnvironment{multline*}{
7687
7688 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7689 {
7690     \LWR@endhideamsmath
7691     \boolfalse{LWR@amsmultline}
7692     \LWR@addmathjax{multline*}{\the\@envbody}
7693 }
7694 {\end{lateximage}}
7695
7696 }
7697

```

Env **gather**

```

7698 \BeforeBeginEnvironment{gather}{
7699
7700 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7701 {
7702     \LWR@syncmathjax
7703     \boolfalse{LWR@amsmultline}
7704     \LWR@beginhideamsmath
7705 }
7706 {
7707     \begin{lateximage}[\LWR@amsmathbodynumbered{gather}]
7708 }
7709 }
7710
7711 \AfterEndEnvironment{gather}{
7712
7713 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7714 {
7715     \LWR@endhideamsmath
7716     \LWR@addmathjax{gather}{\the\@envbody}
7717 }
7718 {\end{lateximage}}
7719
7720 }

```

Env **gather***

```

7721 \BeforeBeginEnvironment{gather*}{
7722
7723 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7724 {
7725     \LWR@syncmathjax
7726     \boolfalse{LWR@amsmultline}
7727     \LWR@beginhideamsmath

```

```

7728 }
7729 {
7730     \begin{lateximage}[\LWR@amsmathbody{gather*}]
7731 }
7732 }
7733
7734 \AfterEndEnvironment{gather*}{
7735
7736 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7737 {
7738     \LWR@endhideamsmath
7739     \LWR@addmathjax{gather*}{\the\@envbody}
7740 }
7741 {\end{lateximage}}
7742
7743 }

```

Env align

```

7744 \BeforeBeginEnvironment{align}{
7745
7746 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7747 {
7748     \LWR@syncmathjax
7749     \boolfalse{LWR@amsmultline}
7750     \LWR@beginhideamsmath
7751 }
7752 {
7753     \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
7754 }
7755 }
7756
7757 \AfterEndEnvironment{align}{
7758
7759 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7760 {
7761     \LWR@endhideamsmath
7762     \LWR@addmathjax{align}{\the\@envbody}
7763 }
7764 {\end{lateximage}}
7765
7766 }

```

Env align*

```

7767 \BeforeBeginEnvironment{align*}{
7768
7769 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%

```

```

7770 {
7771   \LWR@syncmathjax
7772   \boolfalse{LWR@amsmultline}
7773   \LWR@beginhideamsmath
7774 }
7775 {
7776   \begin{lateximage}[\LWR@amsmathbody{align*}]
7777 }
7778 }
7779
7780 \AfterEndEnvironment{align*}{
7781
7782 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7783 {
7784   \LWR@endhideamsmath
7785   \LWR@addmathjax{align*}{\the\@envbody}
7786 }
7787 {\end{lateximage}}
7788
7789 }

```

Env **flalign**

```

7790 \BeforeBeginEnvironment{flalign}{
7791
7792 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7793 {
7794   \LWR@syncmathjax
7795   \boolfalse{LWR@amsmultline}
7796   \LWR@beginhideamsmath
7797 }
7798 {
7799   \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
7800 }
7801 }
7802
7803 \AfterEndEnvironment{flalign}{
7804
7805 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7806 {
7807   \LWR@endhideamsmath
7808   \LWR@addmathjax{flalign}{\the\@envbody}
7809 }
7810 {\end{lateximage}}
7811
7812 }

```

Env `flalign*`

```

7813 \BeforeBeginEnvironment{flalign*}{
7814
7815 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7816 {
7817     \LWR@syncmathjax
7818     \boolfalse{LWR@amsmultline}
7819     \LWR@beginhideamsmath
7820 }
7821 {
7822     \begin{lateximage}[\LWR@amsmathbody{flalign*}]
7823 }
7824 }
7825
7826 \AfterEndEnvironment{flalign*}{
7827
7828 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
7829 {
7830     \LWR@endhideamsmath
7831     \LWR@addmathjax{flalign*}{\the\@envbody}
7832 }
7833 {\end{lateximage}}
7834
7835 }

7836 \end{warpHTML}

```

71 Lateximages

71.1 Description

Env `lateximage` A `lateximage` is a piece of the document which is typeset in \LaTeX then included in the HTML output as an image. This is used for math if `svg math` is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 69.)

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 \TeX 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}`

71.2 Support counters and macros

for HTML output: 7837 `\begin{warpHTML}`

Ctr LWR@lateximagenumber Sequence the images.

```
7838 \newcounter{LWR@lateximagenumber}
7839 \setcounter{LWR@lateximagenumber}{0}
```

Ctr LWR@lateximagedepth Do not create `\lateximage` inside of `\lateximage`.

```
7840 \newcounter{LWR@lateximagedepth}
7841 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
7842 \edef\LWR@hashmark{\string#} % for use in \write
7843 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctr LWR@LIpage Used to reference the PDF page number of a lateximage to be written into `lateximages.txt`.


```
7844 \newcounter{LWR@LIpage}
```

```
7845 \end{warpHTML}
```

71.3 Font size

for HTML & PRINT: 7846 `\begin{warppall}`

`\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.*

```
7847 \newcommand*{\LateximageFontSizeName}{large}

7848 \end{warpall}
```

71.4 Sanitizing math expressions for HTML

for HTML output: 7849 \begin{warpHTML}

\LWR@HTMLsanitize {$text$}

Math expressions are converted to lateximages, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML <alt> tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```
7850 \newcommand{\LWR@HTMLsanitize}[1]{%
7851 \begingroup%
7852 \LWR@FBcancel%
7853 \protect\StrSubstitute{\detokenize{#1}}{%
7854 {\detokenize{&}}{%
7855 {\detokenize{&}}[\LWR@strresult]}%
7856 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
7857 {\detokenize{<}}{%
7858 {\detokenize{<}}[\LWR@strresult]}%
7859 [\LWR@strresult]}%
7860 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
7861 {\detokenize{>}}{%
7862 {\detokenize{>}}[\LWR@strresult]}%
7863 [\LWR@strresult]}%
7864 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}{%
7865 {\detokenize{##}}{%
7866 {\detokenize{##}}[\LWR@strresult]}%
7867 [\LWR@strresult]}%
7868 \LWR@strresult%
7869 \endgroup%
7870 }
```

\LWR@HTMLsanitizeexpand {$text$}

This version expands the argument before sanitizing it.

```
7871 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
```

```

7872 \begingroup%
7873 \LWR@FBcancel%
7874 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
7875 {\detokenize{&}}}%
7876 {\detokenize{&}}}%
7877 [\LWR@strresult]%
7878 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7879 {\detokenize{<}}}%
7880 {\detokenize{<}}}%
7881 [\LWR@strresult]%
7882 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
7883 {\detokenize{>}}}%
7884 {\detokenize{>}}}%
7885 [\LWR@strresult]%
7886 \LWR@strresult%
7887 \endgroup%
7888 }

```

71.5 Equation numbers

Ctr **LWR@startingequation** For use with lateximage and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```

7889 \newcounter{LWR@startingequation}
7890
7891 \@ifundefined{chapter}
7892 {
7893 \renewcommand{\theLWR@startingequation}{%
7894 \arabic{LWR@startingequation}%
7895 }
7896 }
7897 {% chapter defined
7898 \renewcommand{\theLWR@startingequation}{%
7899 \ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}}}%
7900 \arabic{LWR@startingequation}%
7901 }
7902 }

```

Bool **LWR@isstartingequation** True for the first equation tag, false for later tags in the same environment.

```

7903 \newbool{LWR@isstartingequation}

```

\LWR@startingequationtag Prints the starting equation number or tag.

```

7904 \let\LWR@startingequationtag\theLWR@startingequation

```

`\LWR@equationtag` Prints the ending equation number or tag.

```
7905 \let\LWR@equationtag\theequation
```

Only if svg math, patch `\tag` after packages have loaded, in case someone else modified `\tag`.

```
7906 \AtBeginDocument{
7907
7908 \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.

```
7909 \NewDocumentCommand{\LWR@remembertag}{m}{%
7910 \ifbool{LWR@isstartingequation}%
7911 {%
7912   \global\boolfalse{LWR@isstartingequation}%
7913   \xdef\LWR@startingequationtag{#1}%
7914 }%
7915 }%
7916 \xdef\LWR@equationtag{#1}%
7917 }%
```

Patches for \mathcal{MS} math `\tag` macro to remember the first tag:

```
7918 \LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
7919 \LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@
7920
7921 \renewcommand*\make@df@tag@@[1]{%
7922 \LWR@remembertag{#1}%
7923 \LWR@origmake@df@tag@@{#1}%
7924 }
7925
7926 \renewcommand*\make@df@tag@@@[1]{%
7927 \LWR@remembertag{#1}%
7928 \LWR@origmake@df@tag@@@{#1}%
7929 }
7930
7931 }% not mathjax
7932 }% AtBeginDocument
```

71.6 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.

```
7933 \newcommand*{\LWR@amsmathbody}[1]
7934 {%
7935 \textbackslash\begin\}\{#1\} %
7936 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
7937 \textbackslash\end\}\{#1\}%
7938 }
```

`\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.

```
7939 \newcommand*{\LWR@amsmathbodynumbered}[1]
7940 {%
7941 \ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}%
7942 {(\LWR@equationtag)}%
7943 {(\LWR@startingequationtag--\LWR@equationtag)} %
7944 \LWR@amsmathbody{#1} %
7945 }
```

71.7 lateximage

Env `lateximage` [$\langle alt \rangle$ tag] [$\langle CSS \text{ style} \rangle$]

```
7946 \catcode'\$=\active%
7947
7948 \NewDocumentEnvironment{lateximage}{0{image}0{}}
7949 {%
7950 \LWR@traceinfo{lateximage: starting on \jobname.pdf page \arabic{page}}%
```

Nested `lateximages` remain one large `lateximage`:

```
7951 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing `lateximage`, simply record one more level:

```
7952 {%
7953 \addtocounter{LWR@lateximagedepth}{1}%
7954 }%
```

Otherwise, this is the outer-most lateximage:

```
7955 {% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
7956 \setcounter{LWR@startingequation}{\value{equation}}%
7957 \addtocounter{LWR@startingequation}{1}%
7958 \booltrue{LWR@isstartingequation}%
7959 \let\LWR@startingequationtag\theLWR@startingequation%
7960 \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
7961 \addtocounter{LWR@lateximagenumber}{1}%
7962 \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
7963 \boolfalse{mathjax}%
```

Be sure that are doing a paragraph:

```
7964 \LWR@ensuredoingapar%
```

Next file:

```
7965 \addtocounter{LWR@externalfilecnt}{1}%
7966 \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be. `\setcounterpageref` assigns `LWR@LIpage` to the page number for the reference `LWRlateximageXXX`:

```
7967 \setcounterpageref{LWR@LIpage}{LWRlateximage\arabic{LWR@lateximagenumber}}%
7968 \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:

```
7969 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
7970 class="lateximagesource"}%
```

Write instructions to the `lateximages.txt` file:

```
7971 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
7972 \immediate\write\LWR@lateximagesfile%
```

```

7973      |\arabic{LWR@LPage}|\arabic{LWR@externalfilecnt}}|
7974      }%

```

Place an open comment tag. This will hide any traces of the lateximage PDF page which were picked up by pdftotext.

```

7975      \LWR@traceinfo{lateximage: about to create open comment}%
7976      \LWR@htmlopencomment%

```

One level deeper:

```

7977      \addtocounter{LWR@lateximagedepth}{1}%

```

Start the new PDF page:

```

7978      \LWR@traceinfo{lateximage: about to create a new page}%
7979      \LWR@orignewpage%

```

Typeset the image in a “standard” width page and font size:

```

7980      \LWR@traceinfo{lateximage: about to create minipage}%
7981      \LWR@origminipage{6in}%
7982      \csuse{LWR@orig\LateximageFontSizeName}%

```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for \hspace, etc. inside a lateximage.

```

7983      \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
7984      \LWR@restoreorigformatting%

```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```

7985      \def\@mpfn{footnote}%
7986      \def\thempfn{\thefootnote}%
7987      \let\@footnotetext\LWR@footnotetext%

```

Create the LWRlateximage<number> label:

```

7988      \LWR@traceinfo{lateximage: about to create label}%
7989      \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
7990      \LWR@traceinfo{lateximage: finished creating the label}%

```

Enable print-mode math functions:

```

7991      \LetLtxMacro$\LWR@origdollar%
7992      \catcode'\$=3% math shift

```

```

7993 \LetLtxMacro\(\LWR@origopenparen%
7994 \LetLtxMacro\)\LWR@origcloseparen%
7995 \LetLtxMacro\[\LWR@origopenbracket%
7996 \LetLtxMacro\]\LWR@origclosebracket%
7997 }% end of outer-most lateximage
7998 \LWR@traceinfo{lateximage: finished start of environment}%
7999 }% end of \begin{lateximage}

```

`\endlateximage` When the environment closes:

```

8000 {% start of \end{lateximage}
8001 \LWR@traceinfo{lateximage: starting end of environment}%

```

Nested more than one deep?

```

8002 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{1}}%

```

If nesting inside an already-existing lateximage, simply record one more level:

```

8003 {\addtocounter{LWR@lateximagedepth}{-1}}%

```

If this is the outer-most lateximage:

```

8004 {% end of outer-most lateximage

```

Finish the lateximage minipage and start a new PDF page:

```

8005 \LWR@origendminipage%
8006 \LWR@orignewpage%
8007 \LWR@origscriptsize%

```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```

8008 \LWR@origvspace*{.5\baselineskip}%
8009 \LWR@htmlclosecomment%
8010 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%

```

Create a link to the lateximage, allowing its natural height:

```

8011 \LWR@subinlineimage[#1]{lateximage}%
8012 {lateximages\OSPathSymbol{}\LWR@origmbox{lateximage-\theLWR@externalfilecnt}}{svg}{#2}%

```

Be sure that are doing a paragraph:

```

8013 \LWR@ensuredoingapar%

```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```
8014 \LWR@htmltag{/span}%
8015 \ifbool{HTMLDebugComments}{%
8016 \LWR@htmlcomment{End of lateximage}%
8017 }{}%
8018 % \LWR@orignewline% Removed to prevent extra space.
```

Undo one lateximage level:

```
8019 \addtocounter{LWR@lateximagedepth}{-1}%
8020 }% end of outer-most lateximage
8021 \LWR@traceinfo{lateximage: done}%
8022 }%
8023 \catcode'\$=3% math shift
8024 \end{warpHTML}
```

for PRINT output: 8025 \begin{warpprint}

Env lateximage [$\langle alt \rangle$ tag] [$\langle CSS \text{ style} \rangle$]

varwidth is used to create a box of the natural width of its contents.

```
8026 \NewDocumentEnvironment{lateximage}{o o}
8027 {\begin{varwidth}[b]{\linewidth}}
8028 {\end{varwidth}}

8029 \end{warpprint}
```

72 center, flushleft, flushright

for HTML output: 8030 \begin{warpHTML}

Env center Replace center functionality with css tags:

```
8031 \renewenvironment*{center}
8032 {
8033 \LWR@forcenewpage
8034 \ifbool{FormatWP}
8035 {\BlockClass[\LWR@origmbox{text-align:center}]{center}}
8036 {\BlockClass{center}}
8037 }
8038 {\endBlockClass}
```

Env `flushright`

```
8039 \renewenvironment*{flushright}
8040 {
8041 \LWR@forcenewpage
8042 \ifbool{FormatWP}
8043 {\BlockClass[\LWR@origmbox{text-align:right}]{flushright}}
8044 {\BlockClass{flushright}}
8045 }
8046 {\endBlockClass}
```

Env `flushleft`

```
8047 \renewenvironment*{flushleft}
8048 {
8049 \LWR@forcenewpage
8050 \ifbool{FormatWP}
8051 {\BlockClass[\LWR@origmbox{text-align:left}]{flushleft}}
8052 {\BlockClass{flushleft}}
8053 }
8054 {\endBlockClass}
```

`\leftline` $\{\langle text \rangle\}$

```
8055 \renewcommand{\leftline}[1]{\begin{flushleft}#1\end{flushleft}}
```

`\centerline` $\{\langle text \rangle\}$

```
8056 \renewcommand{\centerline}[1]{\begin{center}#1\end{center}}
```

`\rightline` $\{\langle text \rangle\}$

```
8057 \renewcommand{\rightline}[1]{\begin{flushright}#1\end{flushright}}
```

```
8058 \end{warpHTML}
```

73 Pre-loaded packages

for HTML output: 8059 \begin{warpHTML}

If textcomp was loaded before lwarp, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```

8060 \@ifpackageloaded{textcomp}
8061 {
8062 \LWR@origRequirePackage{lwarp-textcomp}
8063 }
8064 {}

```

If graphics or graphicx were loaded before lwarp, perhaps by xunicode, explicitly load the lwarp patches now:

```

8065 \@ifpackageloaded{graphics}
8066 {
8067 \LWR@origRequirePackage{lwarp-graphics}
8068 }
8069 {}

```

```

8070 \end{warpHTML}

```

74 Siunitx

Pkg **siunitx** The lwarp core passes a few options to siunitx.

fractions Due to pdftolatex limitations, fraction output is replaced by symbol output for per-mode and quotient-mode.

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

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

for HTML output: 8071 \begin{warpHTML}

Options for siunitx:

```

8072 \PassOptionsToPackage{
8073     detect-mode=true,
8074     per-mode=symbol,% fraction is not seen by pdftotext
8075 %     text-celsius = {\protect\LWRsiunitx@degree}{C},
8076 %     text-degree = {\protect\LWRsiunitx@degree},
8077 }{siunitx}
8078
8079 \end{warpHTML}

```

75 Graphics print-mode modifications

75.1 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.

Prog `pdftocairo` To convert a PDF image to SVG, use the utility `pdftocairo`:

```
Enter ⇒ pdftocairo -svg filename.pdf
```

If a .pdf file is 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
```

Pkg `epstopdf` For .eps files, use `epstopdf` to provide a PDF version, and also provide a SVG version as well.

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 \TeX , so expect some ugly results for scaling and rotating.

75.2 Print-mode modifications

for PRINT output: For print output, accept and then discard the new `class` key:

```
8080 \begin{warpprint}
8081 \define@key{Gin}{class}{}

```

Print-mode additions for the `overpic` package. See section 237 for the HTML version.

```
8082 \AtBeginDocument{
8083 \ifpackageloaded{overpic}{
8084 \newcommand*{\overpicfontsize}{12}
8085 \newcommand*{\overpicfontskip}{14}
8086 }{}
8087 }
8088 \end{warpprint}

```

76 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` definitions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined `\AtBeginDocument` so that the `xcolor` originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of none, in which case only the frame is drawn, allowing the background page color to show.

for HTML & PRINT: 8089 `\begin{warpall}`

After `xparse` may have been loaded ...

```
8090 \AtBeginDocument{

```

... and *only* if `xcolor` was loaded:

```
8091 \ifpackageloaded{xcolor}{

```

```
8092 \LWR@traceinfo{patching xcolor}
```

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

```
8093 \LetLtxMacro\colorboxBlock\colorbox
```

In HTML mode, the following is done when `xcolor` is loaded. Following is the print-mode action:

```
8094 \warpprintonly{
8095 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
8096 \LetLtxMacro\LWRorigprint@fcolorbox\colorbox
8097 \LetLtxMacro\LWRorigprint@fcolorboxBlock\colorbox
8098 }
```

`\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`.)

```
8099 \newsavebox{\LWR@colorminipagebox}
8100
8101 \DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%
8102 \LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
8103 \begin{lrbox}{\LWR@colorminipagebox}%
8104 #5%
8105 \end{lrbox}%
```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```
8106 \ifthenelse{\equal{#4}{none}}{%
8107 {% #4 none
8108   \LWR@traceinfo{background is none}%
8109   {% scope the \colorlet
8110     \colorlet{\LWR@currentcolor}{.}%
8111     \color{#2}%
8112     \fbox{%
8113       \color{\LWR@currentcolor}%
```

```

8114          \usebox{\LWR@colorminipagebox}%
8115      }% fbox
8116  }% colorlet
8117 }% #4 none
8118 {% #4 not none
8119 \LWR@traceinfo{background not none}%
8120 \IfValueTF{#1}%
8121 {%
8122     \IfValueTF{#3}%
8123     {\LWRorigprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
8124     {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8125 }%
8126 {% no value #1
8127     \IfValueTF{#3}%
8128     {\LWRorigprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
8129     {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8130 }% no value #1
8131 }% #4 not none
8132 \LWR@traceinfo{\LWRprint@fcolorbox done}%
8133 }

```

`\fcolorboxBlock` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```
8134 \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:

```

8135 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
8136 {%
8137 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
8138 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```

8139 \IfValueTF{#7}%
8140 {\begin{minipage}[#5][#6][#7]{#8}}%
8141 {\begin{minipage}[#5][#6][#5]{#8}}%
8142 }%
8143 {%
8144 \end{minipage}%

```

```

8145 \end{lrbox}%
8146 \LWR@traceinfo{*** starting end fcolorminipage #1 #2 #3 #4 #8}%

```

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a \fcolorbox.

The current color is remembered, then set to the frame, then the current color is used for the contents.

```

8147 \ifthenelse{\equal{#4}{none}}%
8148 {% #4 none
8149   {% scope the \colorlet
8150     \colorlet{LWR@currentcolor}{.}%
8151     \color{#2}%
8152     \fbox{%
8153       \color{LWR@currentcolor}%
8154       \usebox{\LWR@colorminipagebox}%
8155     }% fbox
8156   }% colorlet
8157 }% #4 none
8158 {% #4 not none
8159   \IfValueTF{#1}%
8160   {%
8161     \IfValueTF{#3}%
8162     {\LWR@origprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
8163     {\LWR@origprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8164   }%
8165   {% no value #1
8166     \IfValueTF{#3}%
8167     {\LWR@origprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
8168     {\LWR@origprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
8169   }% no value #1
8170 }% #4 not none
8171 \LWR@traceinfo{*** finished end fcolorminipage}%
8172 }

```

`\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`.

```

8173 \newcommand*{\LWR@restoreorigprintxcolor}{%
8174 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
8175 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
8176 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
8177 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
8178 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
8179 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
8180 }
8181
8182 \appto{\LWR@restoreorigformatting}{%

```

```
8183 \LWR@restoreorigprintxcolor%
8184 }
```


If print mode, immediately activate the print-mode enhancements for xcolor:

```
8185 \warpprintonly{\LWR@restoreorigprintxcolor}
8186
8187 \LWR@traceinfo{xcolor patches done}
8188 }{}% xcolor loaded
8189 }% AtBeginDocument

8190 \end{warppall}
```

77 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 section 77 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 373 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: 8191 \begin{warppHTML}

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
8192 \AtEndPreamble{
8193 \RequirePackage{cleveref}
8194 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`. Print-mode versions are not required since they all come down to `\ref` eventually, and `\ref` has a print-mode version.

```
8195 \AfterEndPreamble{
8196 \LWR@traceinfo{Patching cleveref.}
```

`\@@@setcref` `{\<kindofref>} {\<label>}`

`\@templabel` becomes the section number.

```
8197 \def\LWR@orig@@@setcref#1#2{\cref@getlabel{#2}{\@templabel}#1{\@templabel}{}}%
8198
8199 \ifdefequal{\@@@setcref}{\LWR@orig@@@setcref}{% before v0.21
8200   \renewcommand*\@@@setcref}[2]{#1{\ref{#2}}{}}
8201 }{
8202   \ifdefequal{\@@@setcref}{\LWR@orig@@@setcref}{% as of v0.21
8203     \renewcommand*\@@@setcref}[2]{#1{\ref{#2}}{}}
8204   }{
8205     \PackageWarning{lwarp-cleveref}{
8206       Unknown verison of cleveref.
8207       \protect\cref\space will fail.
8208     }%
8209   }
8210 }
```

`\@@@setcrefrange` `{\<text>} {\<label>} {\<label>}`

```
8211 \def\LWR@orig@@@setcrefrange#1#2#3{%
8212   \cref@getlabel{#2}{\@labela}%
8213   \cref@getlabel{#3}{\@labelb}%
8214   #1{\@labela}{\@labelb}{}}%
8215
8216 \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{
8217   \renewcommand{\@@@setcrefrange}[3]{%
8218     #1{\ref{#2}}{\ref{#3}}{}}%
8219 }
8220 }{
8221   \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{
8222     \renewcommand{\@@@setcrefrange}[3]{%
```

```

8223         #1{\ref{#2}}{\ref{#3}}{}{}{}{}%
8224     }
8225 }{
8226     \PackageWarning{lwarp-cleveref}{
8227         Unknown verison of cleveref.
8228         \protect\crefrange\space will fail.
8229     }
8230 }
8231 }
8232

```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```

8233 \newcommand*{\cpagerefFor}{for}

```

`\@@@setcpageref` $\langle typeofref \rangle \langle label \rangle$, where *typeofref* is “page” or “pages”

```

8234 \def\LWR@orig@@setcpageref#1#2{% before v0.21
8235     \cref@getpageref{#2}{\@temppage}#1{\@temppage}{}{}{}%
8236
8237 \def\LWR@orig@@@setcpageref#1#2{% as of v0.21
8238     \cpageref@getlabel{#2}{\@temppage}#1{\@temppage}{}{}{}%
8239
8240 \ifdefequal{\@@setcpageref}{\LWR@orig@@setcpageref}{
8241     \renewcommand*{\@@setcpageref}[2]{%
8242         #1{\cpagerefFor\ \cref{#2}}{}{}{}%
8243     }
8244 }{
8245     \ifdefequal{\@@@setcpageref}{\LWR@orig@@@setcpageref}{
8246         \renewcommand*{\@@@setcpageref}[2]{%
8247             #1{\cpagerefFor\ \cref{#2}}{}{}{}%
8248         }
8249     }
8250     {
8251         \PackageWarning{lwarp-cleveref}{
8252             Unknown verison of cleveref.
8253             \protect\cpageref\space will fail.
8254         }
8255     }
8256 }

8257 \def\LWR@orig@@setcpagerefrange#1#2#3{% before v0.21
8258     \cref@getpageref{#2}{\@pagea}%
8259     \cref@getpageref{#3}{\@pageb}%
8260     #1{\@pagea}{\@pageb}{}{}{}{}{}%
8261
8262 \def\LWR@orig@@@setcpagerefrange#1#2#3{% as of v0.21

```

```

8263 \cpageref@getlabel{#2}{\@pagea}%
8264 \cpageref@getlabel{#3}{\@pageb}%
8265 #1{\@pagea}{\@pageb}{-}{-}{-}{-}%
8266
8267 \ifdefequal{\@@setcpagerefrange}{\LWR@orig@@setcpagerefrange}{
8268   \renewcommand*{\@@setcpagerefrange}[3]{%
8269     #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{-}{-}%
8270   }
8271 }{
8272   \ifdefequal{\@@@setcpagerefrange}{\LWR@orig@@@setcpagerefrange}{
8273     \renewcommand*{\@@@setcpagerefrange}[3]{%
8274       #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{-}{-}%
8275     }
8276   }
8277   {
8278     \PackageWarning{lwarp-cleveref}{
8279       Unknown verison of cleveref.
8280       \protect\cpagerefrange\space will fail.
8281     }
8282   }
8283 }
8284
8285 }% AfterEndPreamble

```

Remember and patch some label-related defintions. These will be further encased and patched by other packages later.

```

8286 \LetLtxMacro\LWR@origlabel\label
8287 \RenewDocumentCommand{\label}{-}{\LWR@newlabel}
8288
8289 \LetLtxMacro\LWR@origref\ref
8290 \RenewDocumentCommand{\ref}{-}{\LWR@newref}%
8291
8292 \LetLtxMacro\LWR@origpageref\pageref
8293 \RenewDocumentCommand{\pageref}{-}{\LWR@newpageref}
8294
8295 \end{warpHTML}

```

78 Picture

Env `picture` The picture environment is enclosed inside a `\lateximage`.

for HTML output: 8296 `\begin{warpHTML}`

Env `picture`

```


8297 \BeforeBeginEnvironment{picture}{\begin{lateximage}}
8298
8299 \AfterEndEnvironment{picture}{\end{lateximage}}

8300 \end{wrapHTML}

```

79 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

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


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

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

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

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

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

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

for HTML output: `8301 \begin{wrapHTML}`

79.1 Counters and lengths

Ctrl `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
8302 \newcounter{LWR@minipagedepth}
8303 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
8304 \newlength{\LWR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
8305 \newlength{\LWR@minipageheight}
```

79.2 Footnote handling

Also see section 50 for other forms of footnotes. Minipage footnotes are gathered in section 50.5, and then placed into the document in section 79.3.

79.3 Minipage handling

Bool `LWR@minipagefullwidth` Should the next minipage have no HTML width?

```
8306 \newbool{LWR@minipagefullwidth}
8307 \boolfalse{LWR@minipagefullwidth}
```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output:

```
8308 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
8309 \end{warpHTML}
```

for PRINT output:

```
8310 \begin{warpprint}
8311 \newcommand*{\minipagefullwidth}{}
8312 \end{warpprint}
```

for HTML output:

```
8313 \begin{warpHTML}
```

Bool `LWR@minipagethispar` Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
8314 \newbool{LWR@minipagethispar}
8315 \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.

```
8316 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
8317 {%
```

Temporarily open a group, in which width and height is computed based on a virtual page size instead of the extra-large PDF page used during HTML tag generation.

The following used to be an actual \LaTeX minipage.

```
8318 \begingroup
```

Compute width, adjusted for frames:

```
8319 \setlength{\LWR@minipagewidth}{#4}%
8320 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%
```

Only create a new page if not yet nested:

```
8321 \LWR@orignewpage%
```

Adjust virtual page size:

```
8322 \addtolength{\LWR@minipagewidth}{3em}% room for frames
8323 \setlength{\linewidth}{6in}%
8324 \setlength{\textwidth}{6in}%
8325 \setlength{\textheight}{9in}%
8326 }{}%
8327 \LWR@traceinfo{computed width is \LWR@printlength{\LWR@minipagewidth}}%
```

Compute height:

```
8328 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
8329 \IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%
```

Track nesting depth:

```
8330 \addtocounter{LWR@minipagedepth}{1}%
```

\LaTeX wants to start a paragraph for the virtual minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

```
8331 \ifbool{FormatWP}{\newline}{}%
8332 \LWR@stoppars%
```

If FormatWP, add a text frame:

```
8333 \ifbool{FormatWP}{%
8334
8335 \addtocounter{LWR@thisautoidWP}{1}%
8336 \LWR@htmltag{div id="\LWR@origmbox{autoidWP-\arabic{LWR@thisautoidWP}}" class="wpminipage"}%
8337
8338 }{}%
```

Create the <div> tag with optional alignment style:

```
8339 \LWR@traceinfo{minipage: creating div class}%
8340 \LWR@htmltag{div class="minipage" style="%
8341 \ifthenelse{\equal{#1}{t}}{\LWR@origmbox{vertical-align:bottom} ; }{}%
8342 \ifthenelse{\equal{#1}{c}}{\LWR@origmbox{vertical-align:middle} ; }{}%
8343 \ifthenelse{\equal{#1}{b}}{\LWR@origmbox{vertical-align:top} ; }{}%
8344 \ifthenelse{\equal{#3}{t}}{\LWR@origmbox{justify-content:flex-start} ; }{}%
8345 \ifthenelse{\equal{#3}{c}}{\LWR@origmbox{justify-content:center} ; }{}%
8346 \ifthenelse{\equal{#3}{b}}{\LWR@origmbox{justify-content:flex-end} ; }{}%
8347 \ifthenelse{\equal{#3}{s}}{\LWR@origmbox{justify-content:space-between} ; }{}%
```

Print the width and optional height styles:

```
8348 \LWR@traceinfo{minipage: about to print the width of \LWR@printlength{\LWR@minipagewidth}}%
8349 \ifbool{LWR@minipagefullwidth}%
8350 {\boolfalse{LWR@minipagefullwidth}}%
8351 {%
8352   \ifthenelse{\lengthtest{#4}=\linewidth}%
8353   {%
8354     {width:\LWR@printlength{\LWR@minipagewidth} ; }%
8355   }%
8356 \LWR@traceinfo{minipage: about to print the height}%
8357 \IfValueT{#2}{height:\LWR@printlength{\LWR@minipageheight} ; }%
8358 "%}
```

Finish with an empty line to start the contents on a new line.

```
8359
8360 % The preceding empty line is required.
```

Set the user-accessible line and text width and height values inside the virtual minipage. These do not affect the actual size of the PDF output, but are used by any reference to \linewidth, etc. inside the virtual minipage being created here.

```
8361 \setlength{\linewidth}{#4}% the original width
```

```
8362 \setlength{\textwidth}{6in}%
8363 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
8364 \LWR@origraggedright%
```

Set minipage footnotes:

```
8365 \def\@mpfn{mpfootnote}%
8366 \def\thempfn{\thempfootnote}\c@mpfootnote\z@%
8367 \let\@footnotetext\@mpfootnotetext%
```

Resume paragraph tag handling for the contents of the minipage:

```
8368 \LWR@startpars%
8369 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
8370
8371 === begin minipage ===
8372
8373 }{}%
8374 \LWR@traceinfo{minipage: finished starting the minipage}%
8375 }% finished \minipage
8376 {% \endminipage
```

Print pending minipage footnotes:

```
8377 \LWR@printpendingmpfootnotes%
```

End the environment with closing tag:

```
8378 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
8379
8380 === end minipage ===
8381
8382 }{}%
8383 \LWR@stoppars%
```

The following used to be an actual \TeX minipage.

```
8384 \endgroup%
8385
8386 \ifbool{FormatWP}{%
8387
8388 \LWR@htmlelementend{div}%
8389
8390 }{}%
8391 \LWR@htmldivclassend{minipage}%
```

```

8392
8393 \addtocounter{LWR@minipagedepth}{-1}%
8394 \LWR@startpars%
8395 \ifbool{FormatWP}{\newline}{}%

```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```

8396 \global\booltrue{LWR@minipagethispar}%
8397 }

```

79.4 Parbox, mbox, makebox, framebox, fbox, raisebox

for HTML output:

```
\parbox [<pos>] [<height>] [<inner-pos>] {<width>} {<text>}
```

A parbox uses the minipage code:

```

8398 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
8399 {
8400 \LWR@traceinfo{parbox of width #4}%
8401 \begin{minipage}[#1][#2][#3]{#4}%
8402 #5
8403 \end{minipage}%
8404 }

```

`\mbox` {*<text>*} Nullified for HTML.

```
8405 \renewcommand*\mbox}[1]{#1}
```

`\makebox` (<()>posn) [*<width>*] [*<pos>*] {*<text>*}

```
8406 \RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```

8407 \IfValueTF{#2}%
8408 {%

```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```

8409 {% scope
8410 \def\LWR@align{center}%
8411 \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%

```

```

8412 \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
8413 \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%

```

To print the width argument:

```

8414 \setlength{\LWR@tempwidth}{#2}%

```

inline-block allows width and text-alignment to be used in a .

```

8415 \InlineClass[%
8416 \LWR@origmbox{display:inline-block} ; %
8417 \LWR@origmbox{text-align}:\LWR@align\ ; %
8418 width:\LWR@printlength{\LWR@tempwidth}%
8419 ]%
8420 {makebox}%

8421 {#4}%
8422 }% scope
8423 }%

```

Without a width argument, the text is simply used inline:

```

8424 {#4}% no width
8425 }

```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

```

8426 \LetLtxMacro\LWR@origframebox\framebox
8427
8428 \RenewDocumentCommand{\framebox}{o o m}{%
8429 \fbox{\makebox[#1][#2]{#3}}%
8430 }

```

`\LWR@forceminwidth` {*<length>*}

Sets `\LWR@atleastonept` to be at least 1pt.

```

8431 \newlength{\LWR@atleastonept}
8432
8433 \newcommand*{\LWR@forceminwidth}[1]{%
8434 \setlength{\LWR@atleastonept}{#1}%
8435 \ifthenelse{%
8436 \lengthtest{\LWR@atleastonept>0pt}\AND%
8437 \lengthtest{\LWR@atleastonept<1pt}}%
8438 }%
8439 {\setlength{\LWR@atleastonept}{1pt}}%
8440 {}%

```

8441 }

`\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.

```
8442 \newcommand*{\LWR@blackborderpadding}{%
8443 border:\LWR@printlength{\LWR@atleastonept} solid black ; %
8444 padding:\LWR@printlength{\fboxsep}%
8445 }
```

`\fbox` $\{\langle text \rangle\}$

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
8446 \let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after xcolor may have loaded:

```
8447 \newcommand{\LWRhtml@fbox}[1]{%
8448 \LWR@traceinfo{HTML fbox}%
8449 \LWR@forceminwidth{\fboxrule}%
8450 \InlineClass[%
8451 \LWR@blackborderpadding%
8452 ]{fbox}{#1}
8453 }
```

`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`:

```
8454 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` $\{\langle text \rangle\}$ Creates a framed HTML `<div>` of the text.

A print-output version is also supplied below.

```
8455 \newcommand{\fboxBlock}[1]{%
8456 \LWR@forceminwidth{\fboxrule}%
8457 \begin{BlockClass}%
8458 \LWR@blackborderpadding%
8459 ]{fboxBlock}
8460 #1
8461 \end{BlockClass}
8462 }
```

Env `fminipage` [*<align>*] [*<height>*] [*<align>*] [*<width>*]

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```

8463 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
8464 {%
8465 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
8466 \LWR@forceminwidth{\fboxrule}%
8467 \setlength{\LWR@tempwidth}{#4}%
8468 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
8469 \begin{BlockClass}{%
8470 \LWR@blackborderpadding ; %
8471 \IfValueT{#2}{height:\LWR@printlength{\LWR@tempheight} ; }%
8472 width:\LWR@printlength{\LWR@tempwidth}%
8473 }{fminipage}%
8474 }
8475 {%
8476 \end{BlockClass}%
8477 \LWR@traceinfo{fminipage done}%
8478 }

```

`\raisebox` [*<raiselen>*] [*<height>*] [*<depth>*] [*<text>*]

```

8479 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
8480
8481 \RenewDocumentCommand{\raisebox}{m o o m}{%
8482 #4%
8483 }

8484 \end{warpHTML}

```

for HTML & PRINT: 8485 \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.

```

8486 \newsavebox{\LWR@fminipagebox}
8487
8488 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
8489 {%

```

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:

```
8490 \IfValueTF{#3}%
8491 {\def\LWR@thisalign{#3}}
8492 {\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.

```
8493 \IfValueTF{#2}%
8494 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
8495 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
8496 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
8497 \IfValueTF{#2}%
8498 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
8499 {\minipage[#1]{#4}}%
8500 }
8501 {%
```

Close the inner minipage and the LR box with the contents:

```
8502 \endminipage%
8503 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
8504 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
8505 \endminipage%
8506 }
```

```
8507 \end{warpall}
```

for PRINT output: 8508 \begin{warpprint}

For print output, the following are `\let` to become active.

`\fboxBlock` $\{\langle text \rangle\}$

Creates a framed HTML `<div>` around the text.

```
8509 \let\fboxBlock\fbox
```

Env `fminipage` $[\langle align \rangle] [\langle height \rangle] [\langle align \rangle] \{\langle width \rangle\}$

Creates a frame around its contents.


```
8510 \LetLtxMacro{fminipage}{\LWRprint@fminipage}
```

```
8511 \LetLtxMacro{endfminipage}{\endLWRprint@fminipage}
```

```
8512 \end{warpprint}
```

80 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 44.8.

for HTML output:

```
8513 \begin{warppHTML}
```

`\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.

```
8514 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
```

```
8515 \ifbool{FormatWP}%
```

```
8516 {\LWR@htmlspanclass[#1]{#2}{#3}}%
```

```
8517 {\LWR@htmlspanclass{#2}{#3}}%
```

```
8518 }
```

`\emph` $\{\langle text \rangle\}$

```
8519 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
```

```
8520 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
```

```
8521 \LetLtxMacro{emph}{\LWR@HTMLemph}
```

`\textmd {<text>}`

```
8522 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
8523 \LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}%
8524 }
8525 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
8526
8527 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

`\textbf {<text>}`

```
8528 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
8529 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
8530 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm {<text>}`

```
8531 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
8532 \LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}%
8533 }
8534
8535 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
8536
8537 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf {<text>}`

```
8538 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
8539 \LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
8540 }
8541 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
8542 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt {<text>}`

```
8543 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
8544 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
8545 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup {<text>}`

```
8546 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
8547 \LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
8548 }
8549
8550 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
```

```
8551
8552 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit` $\{\langle text \rangle\}$

```
8553 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
8554 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
8555 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc` $\{\langle text \rangle\}$

```
8556 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
8557 \LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
8558 }
8559
8560 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
8561
8562 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

`\textsl` $\{\langle text \rangle\}$

```
8563 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
8564 \LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
8565 }
8566
8567 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
8568
8569 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}
```

`\textnormal` $\{\langle text \rangle\}$

```
8570 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
8571 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
8572 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}
```

```
8573 \DeclareRobustCommand{\LWR@nullrmfamily}{}
8574 \DeclareRobustCommand{\LWR@nullsffamily}{}
8575 \DeclareRobustCommand{\LWR@nullttfamily}{}
8576 \DeclareRobustCommand{\LWR@nullbfseries}{}
8577 \DeclareRobustCommand{\LWR@nullmdseries}{}
8578 \DeclareRobustCommand{\LWR@nullupshape}{}
8579 \DeclareRobustCommand{\LWR@nullslshape}{}
8580 \DeclareRobustCommand{\LWR@nullscshape}{}
8581 \DeclareRobustCommand{\LWR@nullitshape}{}
8582 \DeclareRobustCommand{\LWR@nullem}[1]{}
8583 \DeclareRobustCommand{\LWR@nullnormalfont}{}

```

`\LWR@nullfonts` Removes formatting during filename operations.

```

8584 \newcommand*{\LWR@nullfonts}{%
8585 \LetLtxMacro{\emph}{\LWR@nullemph}%
8586 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
8587 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
8588 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
8589 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
8590 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
8591 \LetLtxMacro{\textup}{\LWR@nulltextup}%
8592 \LetLtxMacro{\textit}{\LWR@nulltextit}%
8593 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
8594 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
8595 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
8596 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
8597 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
8598 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
8599 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
8600 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
8601 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
8602 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
8603 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
8604 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
8605 \LetLtxMacro{\em}{\LWR@nullem}%
8606 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
8607 \renewcommand*{\,}{-}%
8608 \renewcommand*{\~}{-}%
8609 \renewcommand*{\textellipsis}{-}%
8610 \renewcommand*{\HTMLUnicode}[1]{-}%
8611 \renewcommand*{\HTMLentity}[1]{-}%

```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```

8612 \renewcommand*{\&}{and}%
8613 \renewcommand{\textsuperscript}[1]{##1}%
8614 \renewcommand{\textsubscript}[1]{##1}%
8615 \LetLtxMacro\underline\LWR@origunderline%
8616 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
8617 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
8618 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%
8619 \DeclareRobustCommand{\LWR@subsingledollar}[1]{}%
8620 \renewcommand*{\newline}{ }%
8621 }

```

Remembers the current font family, series, and shape.

```

8622 \newcommand*{\LWR@f@family}{rm}
8623 \newcommand*{\LWR@f@series}{md}

```

```
8624 \newcommand*{\LWR@f@shape}{up}
```

```
\LWR@textcurrentfont  {\langle text\rangle}
```

Prints the text with the current font choices.

```
8625 \newcommand*{\LWR@textcurrentfont}[1]{%
8626 \csuse{text\LWR@f@family}{%
8627 \csuse{text\LWR@f@series}{%
8628 \csuse{text\LWR@f@shape}{%
8629 #1%
8630 }%
8631 }%
8632 }%
8633 }
```

```
\mdseries
```

```
8634 \renewcommand*{\mdseries}{\renewcommand*{\LWR@f@series}{md}}
```

```
\bfseries
```

```
8635 \renewcommand*{\bfseries}{\renewcommand*{\LWR@f@series}{bf}}
```

```
\rmfamily
```

```
8636 \renewcommand*{\rmfamily}{\renewcommand*{\LWR@f@family}{rm}}
```

```
\sffamily
```

```
8637 \renewcommand*{\sffamily}{\renewcommand*{\LWR@f@family}{sf}}
```

```
\ttfamily
```

```
8638 \renewcommand*{\ttfamily}{\renewcommand*{\LWR@f@family}{tt}}
```

```
\upshape
```

```
8639 \renewcommand*{\upshape}{\renewcommand*{\LWR@f@shape}{up}}
```

```
\itshape
```

```
8640 \renewcommand*{\itshape}{\renewcommand*{\LWR@f@shape}{it}}
```

`\scshape`

```
8641 \renewcommand*{\scshape}{\renewcommand*{\LWR@f@shape}{sc}}
```

`\normalfont`

```
8642 \renewcommand*{\normalfont}{\rmfamily\mdseries\upshape}
```

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

```
8643 \renewcommand{\sp}[1]{\text{<sup>#1</sup>}{}}
```

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

```
8644 \renewcommand{\sb}[1]{\text{<sub>#1</sub>}{}}
```

`\textsuperscript` $\langle text \rangle$

```
8645 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\@textsuperscript` $\langle text \rangle$

```
8646 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript` $\langle text \rangle$

```
8647 \AtBeginDocument{
8648 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
8649 }
```

`\@textsubscript` $\langle text \rangle$

```
8650 \AtBeginDocument{
8651 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
8652 }
```

`\up` $\langle text \rangle$ Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
8653 \AtBeginDocument{\let\up\textsuperscript}
```

`\fup` $\{\langle text \rangle\}$ Prints superscript.

Supports `fmtcount` package.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
8654 \AtBeginDocument{\let\fup\textsuperscript}
```

`\underline` $\{\langle text \rangle\}$

```
8655 \renewcommand{\underline}[1]{%
8656 \LWR@HTMLtextstyle%
8657   {text-decoration:underline;text-decoration-skip}%
8658   {\underline}{#1}%
8659 }

8660 \end{warpHTML}
```

81 Skips, spaces, font sizes

for HTML output: 8661 `\begin{warpHTML}`

`\,` must be redefined after `\RequirePackage{printlen}`

Direct-formatting space commands become HTML entities:

```
8662 \renewrobustcmd*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space

8663
8664 \renewrobustcmd*{\~}{\HTMLentity{nbsp}}
8665
8666 \renewrobustcmd*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
8667 \renewcommand*{\normalsize}{}
8668 \renewcommand*{\small}{}
8669 \renewcommand*{\footnotesize}{}
8670 \renewcommand*{\scriptsize}{}
8671 \renewcommand*{\tiny}{}
8672 \renewcommand*{\large}{}
8673 \renewcommand*{\Large}{}
8674 \renewcommand*{\LARGE}{}
8675 \renewcommand*{\huge}{}
8676 \renewcommand*{\Huge}{}

```

```

8677 \DeclareDocumentCommand{\onecolumn}{}{\}
8678
8679 \DeclareDocumentCommand{\twocolumn}{0{}}{\}
8680
8681 #1
8682
8683 }

```

`\hfill`

```

8684 \renewcommand*{\hfill}{\quad}

```

`\hrulefill`

```

8685 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}

```

`\dotfill`

```

8686 \renewcommand*{\dotfill}{\dots}

```

`\newpage`

```

8687 \renewcommand*{\newpage}{
8688
8689 }

```

`\newline` Uses the HTML `
` element.

```

8690 \newcommand*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
8691 \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`.

```

8692 \LetLtxMacro\LWR@origendofline\\
8693 \NewDocumentCommand{\LWR@endofline}{s o}
8694 {%
8695 \newline%
8696 }

```

`\LWR@minipagestartpars` Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `<div>` to be inside a `p`, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, `lwarp` correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. `lwarp` tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qquad` until the end of the paragraph, when the closing `p` tag is created.

When a minipage is seen, the boolean `LWR@minipagethispar` is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. `LWR@minipagethispar` is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
8697 \newcommand*{\LWR@minipagestartpars}{%
8698 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
8699 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
8700 \newcommand*{\LWR@minipagestoppars}{%
8701 \ifbool{LWR@minipagethispar}{\LWR@stopars}{}%
8702 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
8703 \renewcommand*{\quad}{%
8704 \LWR@minipagestoppars%
8705 \HTMLUnicode{2001}%
8706 \LWR@minipagestartpars%
8707 }
```

`\qquad` Handles special minipage & horizontal space interactions.

```
8708 \renewcommand*{\qquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
8709 \renewcommand*{\enskip}{%
8710 \LWR@minipagestoppars%
8711 \HTMLUnicode{2000}%
```

```
8712 \LWR@minipagestartpars%
8713 }
```

Len \WR@tempwidth Used to compute span width, height, raise for \hspace and \rule:

```
Len \WR@tempheight 8714 \newlength{\LWR@tempwidth}
Len \WR@tempraise 8715 \newlength{\LWR@tempheight}
8716 \newlength{\LWR@tempraise}
```

\LWR@hspace * {*length*}

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

\hspace{*fill*} is converted to \hspace{2em}, equal to \quad.

```
8717 \NewDocumentCommand{\LWR@hspace}{s m}{%
8718 \setlength{\LWR@tempwidth}{#2}%
```

If *fill*, change to \quad:

```
8719 \ifnum\gluestretchorder\LWR@tempwidth>0%
8720 \setlength{\LWR@tempwidth}{2em}%
8721 \fi%
```

Only if the width is not zero:

```
8722 \ifdimcomp{\LWR@tempwidth}{=}{0pt}{}{}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
8723 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
8724 \ifdimcomp{\LWR@tempwidth}{=}{.16667em}%
8725 {%
8726 \HTMLUnicode{2009}% thin breakable space
8727 }%
```

Print the span with the converted width. Not rounded.

```
8728 {%
8729 \LWR@htmltagc{%
8730 span style="width:\LWR@printlength{\LWR@tempwidth}; %
```

```

8731             display:inline-block"%
8732         }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```

8733         \ifbool{FormatWP}{%
8734             \setlength{\LWR@templengthone}{\LWR@tempwidth}%
8735             \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
8736                 \quad%
8737                 \addtolength{\LWR@templengthone}{-1em}%
8738             }%
8739         }{}%
```

Close the span:

```

8740         \LWR@htmltagc{/span}%
8741     }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

8742     \LWR@minipagestartpars%
8743 }% width not 0
8744 }
```

`\LWR@nohspace` * $\{\langle length \rangle\}$

Used to disable \hspace while creating description \items.

```
8745 \NewDocumentCommand{\LWR@nohspace}{s m}{}
```

`\hspace` * $\{\langle length \rangle\}$

Handles special minipage & horizontal space interactions.

```
8746 \LetLtxMacro{\hspace}{\LWR@hspace}
```

`\LWR@vspace` * $\{\langle length \rangle\}$ Nullified vspace.

```
8747 \NewDocumentCommand{\LWR@vspace}{s m}{}
```

`\vspace` * $\{\langle length \rangle\}$ Nullified.

```
8748 \let\vspace\LWR@vspace
```

`\linebreak` [$\langle num \rangle$] Inserts an HTML br tag.

```
8749 \renewcommand*{\linebreak}[1] [] {\newline}
```

`\nolinebreak` [$\langle num \rangle$]

```
8750 \renewcommand*{\nolinebreak}[1] [] {}
```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
8751 \renewcommand*{\pagebreak}[1] [] {
8752
8753 }
```

`\nopagebreak` [$\langle num \rangle$]

```
8754 \renewcommand*{\nopagebreak}[1] [] {}
```

`\enlargethispage` * $\{\langle len \rangle\}$

```
8755 \RenewDocumentCommand{\enlargethispage}{s m}{}%
```

`\clearpage`

`\cleardoublepage`

```
8756 \renewcommand*{\clearpage}{}
8757 \renewcommand*{\cleardoublepage}{}%
```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
8758 \newcommand*{\LWR@currenttextcolor}{black}
```

`\LWR@rule` [$\langle raise \rangle$] $\{\langle width \rangle\}$ $\{\langle height \rangle\}$

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
8759 \NewDocumentCommand{\LWR@rule}{o m m}{%
%
```

The width is copied into a temporary \TeX length, from which comparisons and conversions may be made:

```
8760 \setlength{\LWR@tempwidth}{#2}%
%
```

If it's zero-width then skip the entire rule:

```
8761 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}
8762 {}% zero- width
8763 {% non-zero width
```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
8764 \ifthenelse{%
8765 \lengthtest{\LWR@tempwidth>0pt}\AND%
8766 \lengthtest{\LWR@tempwidth<1pt}}%
8767 }%
8768 {\setlength{\LWR@tempwidth}{1pt}}}%
```

Likewise with height:

```
8769 \setlength{\LWR@tempheight}{#3}%
8770 \ifthenelse{%
8771 \lengthtest{\LWR@tempheight>0pt}\AND%
8772 \lengthtest{\LWR@tempheight<1pt}}%
8773 }%
8774 {\setlength{\LWR@tempheight}{1pt}}}%
```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
8775 \LWR@minipagestoppars%
```

Print the span with the converted width and height. The width and height are NOT rounded, since a height of less than 1pt is quite common in \LaTeX code.

```
8776 \LWR@htmltagc{%
8777 span
8778 style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
8779 \ifbool{FormatWP}{\background:\LWR@currenttextcolor ; }%
```

The width and height are printed, converted to PT:

```
8780 width:\LWR@printlength{\LWR@tempwidth} ; %
8781 height:\LWR@printlength{\LWR@tempheight} ; %
```

The raise height is converted to a css transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a \LaTeX length allows a typical

\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.

```

8782 \IfValueT{#1}%
8783 {%
8784   \setlength{\LWR@tempraise}{Opt-#1}%
8785   \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
8786   \LWR@orignewline%
8787   -ms-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
8788   \LWR@orignewline%
8789   -webkit-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
8790   \LWR@orignewline%
8791   transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
8792   \LWR@orignewline%
8793 }%
```

Display inline-block to place the span inline with the text:

```

8794 display:inline-block;"%
8795 }%
```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```

8796 \ifbool{FormatWP}{%
8797   \setlength{\LWR@templengthone}{\LWR@tempwidth}%
8798   \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
8799     \_{}%
8800     \addtolength{\LWR@templengthone}{-1em}%
8801   }%
8802 }{}%
```

Close the span:

```

8803 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

8804 \LWR@minipagestartpars%
8805 }% non-zero width
8806 }
```

`\rule` [$\langle raise \rangle$] $\{\langle width \rangle\}$ $\{\langle height \rangle\}$

Handles special minipage & horizontal space interactions.

```
8807 \renewcommand{\rule}{\LWR@rule}
```

```
8808 \end{warpHTML}
```

82 \phantomsection

for HTML output: 8809 \begin{warpHTML}

`\phantomsection` Emulate the hyperref `\phantomsection` command, often used to insert the bibliography into table of contents:

```
8810 \DeclareDocumentCommand{\phantomsection}{}{%
```

```
8811 \section*{}}%
```

```
8812 }
```

```
8813 \end{warpHTML}
```

83 \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>

83.1 HTML logos

for HTML output: 8814 \begin{warpHTML}

`\TeX` $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`latexlogo` is a css class used to properly typeset the E and A in $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and friends.

latexlogofont is a css class used to select the font for the rest of the logo in \LaTeX , \LuaTeX , \ConTeXt , etc.

```

8815 \let\LWR@origTeX\TeX
8816
8817 \newcommand*{\LWR@TeX}
8818 {%
8819   \InlineClass{latexlogofont}%
8820   {%
8821     \LWR@HTMLtextstyle%
8822     {text-transform:uppercase}%
8823     {latexlogo}%
8824     {T\textsubscript{e}X}%
8825   }%
8826 }
```

\LaTeX \LaTeX , $\text{\LaTeX} 2_{\epsilon}$

\LaTeXe

```

8827 \let\LWR@origLaTeX\LaTeX
8828
8829 \newcommand*{\LWR@LaTeX}
8830 {%
8831   \InlineClass{latexlogofont}%
8832   {%
8833     \LWR@HTMLtextstyle%
8834     {text-transform:uppercase}%
8835     {latexlogo}%
8836     {L\textsuperscript{a}T\textsubscript{e}X}%
8837   }%
8838 }
8839
8840 \let\LWR@origLaTeXe\LaTeXe
8841
8842 \renewcommand*{\LaTeXe}
8843 {\LaTeX\InlineClass{latexlogofont}%
8844 {\,2\textsubscript{\textit{\HTMLunicode{3B5}}}}}
```

\LuaTeX \LuaTeX , \LuaTeX

\LuaLaTeX

```

8845 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}\Lua\TeX}
8846 \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 Xe_ℒTeX and spaces ℒ_ℒTeX appropriately.

```
8847 \newcommand*{\Xe}
8848   {X\textsubscript{\HTMLUnicode{18e}}}
8849 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
8850 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}
```

\ConTeXt ConT_ℒXt

```
8851 \newcommand*{\LWR@ConTeXt}
8852 {\InlineClass{latexlogofont}{Con}\TeX{}}%
8853 \InlineClass{latexlogofont}{t}}
```

\BibTeX BIB_ℒTeX, *MakeIndex*
\MakeIndex

```
8854 \providecommand*{\BibTeX}
8855 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
8856
8857 \newcommand*{\MakeIndex}
8858 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}
```

\AmS \mathcal{MS}

amslogo is a CSS class used for the \mathcal{MS} logo.

```
8859 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}}
8860 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}
```

\MiKTeX MiK_ℒTeX

```
8861 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

\LyX LyX

lyxlogo is a CSS class used for the LyXlogo.

```
8862 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
8863 \end{warpHTML}
```

83.2 Print logos

for PRINT output:

```

8864 \begin{warpprint}
8865 \newcommand*{\XeTeXrevE}
8866   {\hspace{- .1667em}\raisebox{- .5ex}{\reflectbox{E}}\hspace{- .125em}}
8867 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
8868 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
8869 \providecommand*{\AmS}{%
8870 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
8871 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
8872 \newcommand*{\LyX}{\textsf{LyX}}
8873 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
8874 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
8875 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
8876 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
8877 \providecommand*{\ConTeXt}{\mbox{Con\TeXt}}
8878 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
8879 \end{warpprint}

```

84 \AtBeginDocument, \AtEndDocument

for HTML output: 8880 \begin{warppHTML}

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the document.

\LWR@LwarpEnd

```

8881 \AfterEndPreamble{\LWR@LwarpStart}
8882 \AtEndDocument{\LWR@LwarpEnd}

8883 \end{warppHTML}

```

85 Koma-script

Load patches to koma-script.

for HTML output: 8884 \begin{warppHTML}

```

8885 \@ifclassloaded{scrbook}{\RequirePackage{lwarp-patch-komascript}}{}
8886 \@ifclassloaded{scrartcl}{\RequirePackage{lwarp-patch-komascript}}{}
8887 \@ifclassloaded{scrreprt}{\RequirePackage{lwarp-patch-komascript}}{}

8888 \end{warppHTML}

```

86 Memoir

Load patches to memoir.

```
for HTML output: 8889 \begin{warpHTML}

8890 \@ifclassloaded{memoir}{\RequirePackage{lwarp-patch-memoir}}{}

8891 \end{warpHTML}
```

87 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-a4.sty**

§ 88 Package **a4**

Pkg a4 a4 is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4}
 2 \newcommand*{\WideMargins}{}

File 3 **lwarp-a4wide.sty**

§ 89 Package **a4wide**

Pkg a4wide a4wide is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4wide}

File 4 **lwarp-a5comb.sty**

§ 90 Package **a5comb**

Pkg a5comb a5comb is ignored.


for HTML output: 1 \LWR@ProvidesPackageDrop{a5comb}

File 5 **lwarp-abstract.sty**

§ 91 Package **abstract**

(Emulates or patches code by PETER WILSON.)

Pkg abstract abstract is supported and patched by lwarp.

 **missing TOC** If using the number option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a

file split, which would put a table of contents out of the home page if it is after the abstract.

for HTML output:

memoir provides an abstract environment even though it is not an article or report class. Meanwhile, lwarp loads book to emulate memoir, but book does not have an abstract environment, so when the abstract package is loaded for emulation there is no pre-existing abstract to redefine, which would cause an error. Thus, a null abstract is provide here:

```
1 \ProvideDocumentEnvironment{abstract}{}{}{}
```

Accept all options for lwarp-abstract:

```
2 \LWR@ProvidesPackagePass{abstract}

3 \AtBeginDocument{
4 \BeforeBeginEnvironment{abstract}{
5 \LWR@forcenewpage
6 \BlockClass{abstract}
7 }
8 \AfterEndEnvironment{abstract}{\endBlockClass}
9 }
10
11 \renewcommand{\@bsrunintitle}{%
12 \hspace*{\abstitlestitle}%
13 {\abstractnamefont%
14 \InlineClass{abstractrunintitle}{\abstractname}%
15 \@bslabeldelim}%
16 }
17
18 \@ifclassloaded{memoir}
19 {
20 \renewenvironment{abstract}{%
21 % \titlepage
22 \null\vfil
23 \@beginparpenalty\@lowpenalty
24 \if@bsrunin
25 \else
26 \if@bsstyle
27 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
28 \else
29 \ifnumber@bs
30 \num@bs
31 \else
32 \begin{\absnamepos}%
33 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
34 \endparpenalty\@M
35 \end\absnamepos%
36 %% \vspace{\abstitlestitle}%
```

```

37      \fi
38      \fi
39      \vspace{\abstitleskip}%
40      \fi
41      \put@bsintoc%
42      \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
43      {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
44      }
45 }{% not memoir
46 \if@titlepage
47   \renewenvironment{abstract}{%
48 %     \titlepage
49     \null\vfil
50     \@beginparpenalty\@lowpenalty
51     \if@bsrunin
52     \else
53       \if@bsstyle
54         \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
55       \else
56         \ifnumber@bs
57           \num@bs
58         \else
59           \begin{\absnamepos}%
60 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
61           \endparpenalty\@M
62           \end\absnamepos%
63 %%       \vspace{\abstitleskip}%
64       \fi
65     \fi
66     \vspace{\abstitleskip}%
67     \fi
68     \put@bsintoc%
69     \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
70     {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
71     }
72 \else
73   \renewenvironment{abstract}{%
74     \if@bsrunin
75     \else
76       \if@bsstyle
77         \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
78       \else
79         \ifnumber@bs
80           \num@bs
81         \else
82 \begin{\absnamepos}%
83 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
84 \end\absnamepos%
85 %%       \vspace{\abstitleskip}%
86     \fi

```

```

87     \fi
88     \vspace{\abstitleskip}%
89     \fi
90     \put@bsintoc%
91     \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
92     {\par\end{@bstr@ctlist}}
93 \fi
94 }% not memoir

```

File 6 lwarp-acro.sty

§ 92 Package **acro**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg acro acro is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{acro}

\DeclareAcronym is used in the preamble, where lwarp has not yet made the dollar active, so temporarily enable lwarp math catcode just for this definition:

```

2 \ExplSyntaxOn
3 \NewDocumentCommand \LWR@DeclareAcronym {mm}
4 {
5     \acro_declare_acronym:nn {#1} {#2}
6     \catcode'\$=3% lwarp
7 }
8 \ExplSyntaxOff
9
10 \RenewDocumentCommand{\DeclareAcronym}{}{}{
11     \catcode'\$=\active% lwarp
12     \LWR@DeclareAcronym
13 }

```

Modified to activate the current font:

```

14 \ExplSyntaxOn
15 \cs_gset_protected:Npn \acro_write_short:nn #1#2
16 {
17     \mode_if_horizontal:F { \leavevmode }
18     \group_begin:
19     \bool_if:NTF \l__acro_custom_format_bool
20         { \l__acro_custom_format_tl }
21         { \l__acro_short_format_tl }
22     {\LWR@textcurrentfont{#2}}% lwarp

```

```

23   \group_end:
24 }
25
26 \cs_gset_protected:Npn \acro_write_alt:nn #1#2
27 {
28   \mode_if_horizontal:F { \leavevmode }
29   \group_begin:
30     \bool_if:NTF \l__acro_custom_format_bool
31       { \l__acro_custom_format_tl }
32       { \l__acro_alt_format_tl }
33     {\LWR@textcurrentfont{#2}}% lwarp
34   \group_end:
35 }
36
37 \cs_gset_protected:Npn \acro_write_long:nn #1#2
38 {
39   \mode_if_horizontal:F { \leavevmode }
40   \group_begin:
41     \bool_if:NTF \l__acro_custom_long_format_bool
42       { \l__acro_custom_long_format_tl }
43       { \use:n }
44     {
45       \use:x
46       {
47         \exp_not:n {#1}
48         {
49           \bool_if:NTF \l__acro_first_upper_bool
50             { \exp_not:N \l__acro_first_upper_case:n { \exp_not:n {
51               \LWR@textcurrentfont{#2}}% lwarp
52             } } }
53           { \exp_not:n {\LWR@textcurrentfont{#2}} }% lwarp
54         }
55       }
56     }
57   \group_end:
58 }
59 \ExplSyntaxOff

```

File 7 **lwarp-acronym.sty**

§ 93 Package **acronym**

(Emulates or patches code by TOBIAS OETIKER.)

Pkg acronym acronym is patched for use by lwarp.

 \acresetall does not work with cleveref, causing multiply-defined labels. lwarp

patches acronym for HTML, but not for print mode.

for HTML output: 1 \LWR@ProvidesPackagePass{acronym}

Uses \textit instead of \itshape:

```
2 \renewcommand{\acfia}[1]{%
3   {\textit{\AC@acl{#1}}}} (\ifAC@starred\acs*{#1}\else\acs{#1}\fi)}
```

Removes the mbox to allow math inside:

```
4 \renewcommand*\AC@acs[1]{%
5   % \mbox{
6   \expandafter\AC@get\csname fn@#1\endcsname\@firstoftwo{#1}}
7 % }
```

Modified for cleveref and zref:

```
8 \renewcommand*\AC@und@newl@bel[3]{%
9   \@ifundefined{#1@#3}%
10  {%
11    \global\expandafter\let\csname#2@#3\endcsname\@nnil
12    \global\expandafter\let\csname#2@#3@cref\endcsname\@nnil% lwarp
13  }%
14  {%
15    \global\expandafter\let\csname#1@#3\endcsname\relax
16    \global\expandafter\let\csname#1@#3@cref\endcsname\relax% lwarp
17    \global\expandafter\let\csname Z@R@#3\endcsname\relax% lwarp
18  }%
19 }%
```

Modified for cleveref and zref:

```
20 \renewcommand*\AC@testdef[3]{%
21 \ifstrequal{#1}{Z@R}{}{% lwarp
22   \@ifundefined{s@#2}\@secondoftwo\@firstofone
23   {%
24     \expandafter\ifx\csname s@#2\endcsname\empty
25     \expandafter\@firstofone
26   \else
27     \expandafter\xdef\csname s@#2\endcsname{%
28       \expandafter\expandafter
29       \expandafter\@gobble
30       \csname s@#2\endcsname
31     }%
32     \expandafter\@gobble
33   \fi
34 }%
35 {%
```

```

36   \@testdef{#1}{#2}{#3}%
37   }%
38 }% lwarp
39 }%

```

File 8 **lwarp-adjmulticol.sty**

\$ 94 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%

```

Create the `<div>` with the given margin and class:

```

12 \BlockClass[%

```

```

13 \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
14 \LWR@origmbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}%
15 ]{\LWR@mcolstype}%
16 }
17 {\endBlockClass}

```

File 9 **lwarp-addlines.sty**

\$ 95 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 10 **lwarp-afterpage.sty**

\$ 96 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 11 **lwarp-algorithmicx.sty**

\$ 97 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 308.1.

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 \LWR@htmltagc{%
10 span style="width:\LWR@printlength{\ALG@thistlm}; display:inline-block;"%
11 }%
12 \ifbool{FormatWP}{%
13 \setlength{\LWR@templengthone}{\the\ALG@thistlm}%
14 \whileof{\lengthtest{\LWR@templengthone>1em}}{%
15 \quad%
16 \addtolength{\LWR@templengthone}{-1em}%
17 }%
18 }{}%
19 \LWR@htmltagc{/span}%
20 }%
21
22 \let\LWR@origComment\Comment%
23
24 \renewcommand{\Comment}[1]{%
25   \InlineClass{floatright}{\LWR@origComment{#1}}%
26 }%
27 }
28
29 \renewcommand\algorithmiccomment[1]{%
30 \hfill\HTMLUnicode{25B7} #1% white right triangle
31 }%

32 \end{warpHTML}
```

File 12 **lwarp-alltt.sty**

§ 98 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 13 lwarp-amsthm.sty

\$ 99 Package amsthm

(Emulates or patches code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

The original source code is located in amsclass.dtx, and printed in amsclass.pdf.

Pkg amsthm amsthm is patched for use by lwarp.

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.

```
for HTML output: 1 \LWR@ProvidesPackagePass{amsthm}

Storage for the style being used for new theorems:

2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```

3 \renewcommand{\theoremstyle}[1]{%
4   \ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
8   }{%
9     \thm@style{#1}%
10    \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
11  }%
12 }

```

Patched to remember the style for this theorem type:

```

13 \def\@xnthm#1#2{%
14   \csedef{LWR@thmstyle#2}{\LWR@newtheoremstyle}% lwarp
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\@xp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
22           }{##1}}}%
23   \else % numbered theorem, need to check for optional arg
24     \def\@tempa{\@oparg{\@ynthm{#2}}{}}%
25   \fi
26   \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% lwarp
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   \item[
47 %   \deferred@thm@head{
48 %     \the\thm@headfont \thm@indent
49   \@ifempty{#1}{\let\thmname\@gobble}{\LWR@haveamsthmname}% lwarp
50   \@ifempty{#2}{\let\thmnumber\@gobble}{\LWR@haveamsthmnumber}% lwarp
51   \@ifempty{#3}{\let\thmnote\@gobble}{\LWR@haveamsthmnote}% lwarp
52   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
53   \the\thm@headpunct~
54   \thmheadnl % possibly a newline.
55   \hskip\thm@headsep
56 %   }%
57   ]
58   \ignorespaces}
```

Patched for CSS:

```
59 \def\@thm#1#2#3{%
60   \ifhmode\unskip\unskip\par\fi
61   \normalfont
62   \LWR@forcenewpage% lwarp
63   \BlockClass{amsthmbody\LWR@thisthmstyle}% lwarp
64   \trivlist
65   \let\thmheadnl\relax
66   \let\thm@swap\@gobble
67   \thm@notefont{\fontseries\mddefault\upshape}%
68   \thm@headpunct{.}% add period after heading
69   \thm@headsep 5\p@ plus\p@ minus\p@\relax
70   \thm@space@setup
71   #1% style overrides
72   \@topsep \thm@preskip           % used by thm head
73   \@topsepadd \thm@postskip       % used by \@endparenv
74   \def\@tempa{#2}\ifx\@empty\@tempa
75     \def\@tempa{\@oparg{\@begintheorem{#3}{}}{}}%
76   \else
77     \refstepcounter{#2}%
78     \def\@tempa{\@oparg{\@begintheorem{#3}{\cename the#2\endcsname}}{}}%
79   \fi
80   \@tempa
81 }
```

cleveref patches \@thm to do \cref@thmoptarg if an optional argument is given.
lwarp then patches \cref@thmoptarg \AtBeginDocument.

```
82 \AtBeginDocument{
```

```

83 \def\cref@thmoptarg[#1]#2#3#4{%
84   \ifhmode\unskip\unskip\par\fi%
85   \normalfont%
86   \LWR@forcenewpage% lwarp
87   \BlockClass{amsthmbody\LWR@thisthmstyle}% lwarp
88   \trivlist%
89   \let\thmheadnl\relax%
90   \let\thm@swap\@gobble%
91   \thm@notefont{\fontseries\mddefault\upshape}%
92   \thm@headpunct{.}% add period after heading
93   \thm@headsep 5\p@ plus\p@ minus\p@\relax%
94   \thm@space@setup%
95   #2% style overrides
96   \@topsep \thm@preskip           % used by thm head
97   \@topsepadd \thm@postskip       % used by \@endparenv
98   \def\@tempa{#3}\ifx\@empty\@tempa%
99     \def\@tempa{\@oparg{\@begintheorem{#4}{}}{}}[]}%
100  \else%
101    \refstepcounter{#1}{#3}% <<< cleveref modification
102    \def\@tempa{\@oparg{\@begintheorem{#4}{\csname the#3\endcsname}}{}}}%
103  \fi%
104  \@tempa
105 }%
106 %% AtBeginDocument
107
108 \def\@endtheorem{\endtrivlist\endBlockClass\@endpefalse }

```

Proof QED symbol:

```

109 \AtBeginDocument{
110 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
111 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
112 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
113 }

```

Patched for CSS:


```

114 \renewenvironment{proof}[1][\proofname]{\par
115 \LWR@forcenewpage% lwarp
116   \BlockClass{amsthmproof}% lwarp
117   \pushQED{\qed}%
118   \normalfont \topsep6\p@\@plus6\p@\relax
119   \trivlist
120   \item[
121     \InlineClass{amsthmproofname}{#1@addpunct{.}}]\ignorespaces% changes
122 ]{%
123   \InlineClass{theoremendmark}{\popQED}\endtrivlist%
124   \endBlockClass% lwarp
125   \@endpefalse

```

126 }

File 14 **lwarp-anonchap.sty**§ 100 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 \LaTeX commands to create the titles, allowing other packages to work with it.

The code is shared by tocbibind.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{anonchap}

2 \newcommand{\simplechapter}[1][\@empty]{%
3   \def\@chapcntformat##1{%
4     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
5   }%
6 }
7
8 \newcommand{\restorechapter}{%
9 \let\@chapcntformat\@secntformat%
10 }
```

File 15 **lwarp-anysize.sty**§ 101 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 16 **lwarp-appendix.sty**

§ 102 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 17 **lwarp-arabicfront.sty**

§ 103 Package **arabicfront**

Pkg arabicfront arabicfront is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{arabicfront}
```

File 18 **lwarp-array.sty**

§ 104 Package **array**

Pkg array array is used as-is for print output, and emulated for HTML.

for HTML output:

```

1 \LWR@ProvidesPackagePass{array}

2 \let\LWR@origfirstline\firstline
3 \let\LWR@origlastline\lastline
4
5 \appto{\LWR@restoreorigformatting}{%
6 \let\firstline\LWR@origfirstline%
7 \let\lastline\LWR@origlastline%
8 }
9
10 \renewcommand*{\firstline}{\LWR@HTMLhline}%
11 \renewcommand*{\lastline}{\LWR@HTMLhline}%

```

File 19 **lwarp-atbegshi.sty**

§ 105 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{0pt}
14 \def\AtBeginShipoutBoxHeight{0pt}
15 \def\AtBeginShipoutBoxDepth{0pt}
16


```

File 20 **lwarp-authblk.sty**

§ 106 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
 load order authblk and titling. If both are used, authblk should be loaded before titling.

\backslash published and \backslash subtitle If using the titling package, additional titlepage fields for \backslash published and \backslash subtitle may be added by using \backslash AddSubtitlePublished in the preamble. See section 57.8.

(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 \backslash authorcr:

```

13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}


```

File 21 **lwarp-backref.sty**

§ 107 Package **backref**

(Emulates or patches code by DAVID CARLISLE AND SEBASTIAN RAHTZ.)

Pkg **backref** backref is patched for use by lwarp.

 **loading** Note that backref must be explicitly loaded, and is not automatically loaded by hyperref when generating HTML output.

for HTML output: 1 \LWR@ProvidesPackagePass{backref}

Force the hyperref option:

2 \def\backref{}\let\backrefxxx\hyper@section@backref

File 22 **lwarp-balance.sty**

§ 108 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 23 **lwarp-bigdelim.sty**

§ 109 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}{l l l}
<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 {<1:delimiter>} {<2:#rows>} {<3:width>} [<4:text>]
\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 24 **lwarp-bigstrut.sty**

§ 110 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 25 **lwarp-blowup.sty**

§ 111 Package **blowup**

Pkg blowup blowup is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{blowup}

2 \newcommand*\blowUp[1]{}

```

File 26 **lwarp-bookmark.sty**

§ 112 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 27 **lwarp-booktabs.sty**

§ 113 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 28 **lwarp-boxedminipage.sty**

§ 114 Package **boxedminipage**

Pkg boxedminipage boxedminipage is superseded by boxedminipage2e.

for HTML output: 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

File 29 **lwarp-boxedminipage2e.sty**

§ 115 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 30 **lwarp-breakurl.sty**

§ 116 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{\burlalt}{0{} +m m}{%
5 \LWR@ensuredoingapar%
6 \def\LWR@templink{#2}%
7 \@onelevel@sanitize\LWR@templink%
8 \def\LWR@templinktwo{#3}%
9 \@onelevel@sanitize\LWR@templinktwo%
10 \LWR@subhyperref{\LWR@templink}{\LWR@templinktwo}%
11 \LWR@ensuredoingapar%
12 }
13
```

```
14 \LetLtxMacro\urlalt\burlalt
```

File 31 **lwarp-cancel.sty**

§ 117 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 32 **lwarp-caption.sty**§ 118 Package **caption***(Emulates or patches code by AXEL SOMMERFELDT.)*

Pkg caption caption is patched for use by lwarp.

```

for HTML output: 1 \LWR@ProvidesPackagePass{caption}

2 \renewcommand\caption@ibox[3]{%
3   \@testopt{\caption@iiibox{#1}{#2}{#3}}{%
4     \wd\@tempboxa%
5     \linewidth% lwarp
6   }%
7 \LWR@traceinfo{caption@ibox: done}%
8 }

9 \long\def\caption@iiibox#1#2#3[#4]{%
10  \@testopt{\caption@iiiibox{#1}{#2}{#3}{#4}}\captionbox@hj@default
11 }

12 \long\def\caption@iiiibox#1#2#3#4[#5]#6{%
13   \setbox\@tempboxa\hbox{#6}%
14   \begingroup
15   #1*% set \caption@position
16   \caption@iftop{%
17     \LWR@traceinfo{caption@iiiibox top}%
18     \endgroup
19     \parbox[t]{#4}{%
20       #1\relax
21       \caption@setposition t%
22 %       \vbox{\caption#2{#3}}%
23       {\caption#2{#3}}% lwarp
24 %       \captionbox@hrule
25 %       \csname caption@hj@#5\endcsname
26 %       \unhbox\@tempboxa
27       #6% lwarp
28     }%
29   }{%
30     \LWR@traceinfo{caption@iiiibox bottom}%
31     \endgroup
32     \parbox[b]{#4}{%
33       #1\relax
34       \caption@setposition b%

```

```

35 %      \csname caption@hj@#5\endcsname
36 %      \unhbox\@tempboxa
37 %      #6% lwarp
38 %      \captionbox@hrule
39 %      \vtop{\caption#2{#3}}}%
40 %      {\caption#2{#3}}% lwarp
41 %      }%
42 %      }%
43 \LWR@traceinfo{caption@iiibox: done}%
44 }
45
46 \def\caption@caption{%
47   \caption@iftype
48   {%
49     \caption@checkgrouplevel\@empty\caption
50     \caption@star
51     {\caption@refstepcounter\@capttype}%
52     {\caption@dblarg{\@caption\@capttype}}}%
53   {\caption@Error{\noexpand\caption outside float}}%
54   \caption@gobble}%
55 }
56
57 \long\def\caption@@caption#1[#2]#3{%

58   \ifcaption@star \else
59     \caption@prepareanchor{#1}{#2}%
60     \memcaptioninfo{#1}{\csname the#1\endcsname}{#2}{#3}%
61     \@nameuse{nag@hascaptiontrue}%
62   \fi

63   \par
64   \caption@beginex{#1}{#2}{#3}%
65   \caption@setfloatcapt{%
66     \caption@boxrestore
67     \if@minipage
68       \@setminipage
69     \fi
70     \caption@normalsize
71     \ifcaption@star
72       \let\caption@makeanchor\@firstofone
73     \fi
74     \@makecaption{\csname fnum@#1\endcsname}%
75     {\ignorespaces\caption@makeanchor{#3}}\par
76     \caption@if@minipage\@minipagetrue\@minipagefalse}%
77   \caption@end%
78 }

```

\caption@@@make {\caption label} {\caption text}

```

79 \renewcommand\caption@@make[2]{%
80 \LWR@startpars% lwarp
81 %   \sbox\@tempboxa{#1}%
82 %   \ifdim\wd\@tempboxa=\z@
83 %     \let\caption@lsep\relax
84 %   \fi
85 \caption@ifempty{#2}{%
86   \let\caption@lsep\@empty
87   \let\caption@tfmt\@firstofone
88 }%
89 \@setpar{\LWR@closeparagraph\@@par}% lwarp
90 \caption@applyfont
91 \caption@fmt
92   {\ifcaption@star\else
93     \begingroup
94       \captionlabelfont
95       #1%
96     \endgroup
97   \fi}%
98   {\ifcaption@star\else
99     \begingroup
100       \caption@iflf\captionlabelfont
101       \relax\caption@lsep
102     \endgroup
103   \fi}%
104   {\caption@textfont
105     \caption@ifstrut
106       {\vrule\@height\ht\strutbox\@width\z@}%
107       {}%
108     \nobreak\hskip\z@skip % enable hyphenation
109     \caption@tfmt{#2}
110   \LWR@ensuredoingapar% lwarp
111   \caption@ifstrut
112     {\ifhmode\@finalstrut\strutbox\fi}%
113     {}%
114   \par}}
115 \LWR@stoppars% lwarp
116 }

```

\caption@@make@ {<>} {<>}

```

117 \renewcommand{\caption@@make@}[2]{%
118   \caption@stepthecounter
119   \caption@beginhook
120   \caption@@make{#1}{#2}%
121   \caption@endhook
122 }

123 % \DeclareCaptionBox{none}{#2}

```

```

124 \DeclareCaptionBox{parbox}{%
125 #2%
126 }
127 \DeclareCaptionBox{colorbox}{%
128 #2%
129 }

```

File 33 **lwarp-caption2.sty**

§ 119 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 34 **lwarp-ccaption.sty**

§ 120 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 35 **lwarp-changebar.sty**

§ 121 Package **changebar**

Pkg changebar changebar is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{changebar}

```

2 \newcommand*{\cbstart}{}
3 \newcommand*{\cbend}{}
4 \newenvironment*{\changebar}{}{}
5 \newcommand*{\cbdelete}{}
6 \newcommand*{\nochnagebars}{}
7 \newcommand*{\cbcolor}[1]{}
8 \newlength{\changebarwidth}
9 \newlength{\deletebarwidth}
10 \newlength{\changebarsep}
11 \newcounter{changebargrey}

```

File 36 **lwarp-changepage.sty**

§ 122 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\ifoddpage
3 \DeclareRobustCommand{\checkoddpage}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\changeage}[9]{}
6
7 \@ifundefined{adjustwidth}{
8 \newenvironment{adjustwidth}[2]{}{}
9 \newenvironment{adjustwidth*}[2]{}{}
10 }{
11 \renewenvironment{adjustwidth}[2]{}{}
12 \renewenvironment{adjustwidth*}[2]{}{}
13 }

14 \DeclareDocumentCommand{\strictpagecheck}{}{}
15 \DeclareDocumentCommand{\easypagecheck}{}{}

```

File 37 **lwarp-chnpage.sty**

§ 123 Package **chnpage**

Pkg chnpage chnpage is superceded by changepage.

for HTML output: 1 \LWR@loadnever{chnpage}{changeage}

File 38 **lwarp-chappg.sty**

§ 124 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 39 **lwarp-chapterbib.sty**

§ 125 Package **chapterbib**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg chapterbib chapterbib is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{chapterbib}

2 \xdef\@savedjobname{\BaseJobname}

3 \let\@currentipfile\@savedjobname

File 40 **lwarp-cite.sty**

§ 126 Package **cite**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg cite cite is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{cite}

For the [super] option, the \kern must be removed:

2 \def\LWRCT@biblabel#1{\@citess{#1}\kern-\labelsep\,}

```

3
4 \ifdefstrequal{\@biblabel}{\LWRCT@biblabel}
5 {
6   \def\@biblabel#1{\@citess{#1}}
7 }{}}

```

For the [super] option, `\textsuperscript` is used instead of math superscript:

```

8 \def\@citess#1{\textsuperscript{#1}}
9
10 \DeclareDocumentCommand\citepunct{}{, \, \relax}


```

File 41 `lwarp-color.sty`

§ 127 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 42 `lwarp-colortbl.sty`

§ 128 Package **colortbl**

Pkg colortbl colortbl is emulated.

 **row/cell color** Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

colortbl ignores the overhang arguments.

for HTML output:

```

1 \LWR@ProvidesPackagePass{colortbl}

```

Remember the print-mode definitions:

```

2 \LetLtxMacro\LWR@origcolumncolor\columncolor
3 \LetLtxMacro\LWR@origrowcolor\rowcolor
4 \LetLtxMacro\LWR@origcellcolor\cellcolor
5 \LetLtxMacro\LWR@origarrayrulecolor\arrayrulecolor
6 \LetLtxMacro\LWR@origdoublerulesepcolor\doublerulesepcolor

```

```

7
8 \appto{\LWR@restoreorigformatting}{%
9 \LetLtxMacro\columncolor\LWR@origcolumncolor
10 \LetLtxMacro\rowcolor\LWR@origrowcolor
11 \LetLtxMacro\cellcolor\LWR@origcellcolor
12 \LetLtxMacro\arrayrulecolor\LWR@origarrayrulecolor
13 \LetLtxMacro\doublerulesepcolor\LWR@origdoublerulesepcolor
14 }

```

The following \LWR@HTML versions are used inside an HTML tabular.

`\columncolor` [*<model>*] {*<color>*} [*<left overhang>*] [*<right overhang>*]

\LWR@getmynexttoken is not used here because \columncolor is not used inside the data area of the tabular.

```

15 \RenewDocumentCommand{\LWR@HTMLcolumncolor}{O{named} m o o}{%
16 \convertcolorspec{#1}{#2}{HTML}\LWR@columnHTMLcolor%
17 \LWR@addtabularcellcolor%
18 }

```

\LWR@getmynexttoken is used for \rowcolor because it is used inside the data area of the tabular.

`\rowcolor` [*<model>*] {*<color>*} [*<left overhang>*] [*<right overhang>*]

```

19 \RenewDocumentCommand{\LWR@HTMLrowcolor}{O{named} m o o}{%
20 \convertcolorspec{#1}{#2}{HTML}\LWR@rowHTMLcolor%
21 \LWR@getmynexttoken%
22 }

```

`\cellcolor` [*<model>*] {*<color>*} [*<left overhang>*] [*<right overhang>*]

```

23 \RenewDocumentCommand{\LWR@HTMLcellcolor}{O{named} m o o}{%
24 \convertcolorspec{#1}{#2}{HTML}\LWR@cellHTMLcolor%
25 \LWR@addtabularcellcolor%
26 }

```

`\arrayrulecolor` [*<model>*] {*<color>*}

The version for use outside a tabular.

```

27 \renewcommand{\arrayrulecolor}[2] [named]{%
28 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
29 }

```

`\LWR@arrayrulecolor` [*<model>*] {*<color>*}

The version for use inside a tabular.

```

30 \renewcommand{\LWR@HTMLarrayrulecolor}[2] [named] {%
31 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
32 \LWR@getmynexttoken%
33 }

```

`\doublerulesepcolor` [*<model>*] {*<color>*}

The version for use outside a tabular.

```

34 \renewcommand{\doublerulesepcolor}[2] [named] {}

```

`\LWR@doublerulesepcolor` [*<model>*] {*<color>*}

The version for use inside a tabular.

```

35 \renewcommand{\LWR@HTMLdoublerulesepcolor}[2] [named] {\LWR@getmynexttoken}

```

File 43 **lwarp-continue.sty**

§ 129 Package **continue**

Pkg `continue` continue is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{continue}

```

2 \newcommand*{\flagcont}{}
3 \newcommand*{\flagend}{}
4 \newcommand*{\flagword}{}
5 \newcommand*{\preflagword}{}
6 \newcommand*{\postflagword}{}
7 \newlength\contsep
8 \newlength\contdrop

```

File 44 **lwarp-crop.sty**

§ 130 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 45 **lwarp-cuted.sty**

§ 131 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 46 **lwarp-cutwin.sty**

§ 132 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 47 **lwarp-dblfnote.sty**

§ 133 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 48 **lwarp-dcolumn.sty**

§ 134 Package **dcolumn**

Pkg dcolumn dcolumn is emulated by the lwarp core.

```

1 \LWR@ProvidesPackageDrop{dcolumn}

```

File 49 **lwarp-draftwatermark.sty**

§ 135 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 50 **lwarp-easy-todo.sty**

§ 136 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 51 **lwarp-ebook.sty**

§ 137 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 52 **lwarp-ellipsis.sty**

§ 138 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 53 **lwarp-emptypage.sty**

\$ 139 Package **emptypage**

Pkg emptypage emptypage is ignored.

for HTML output: Discard all options for lwarp-emptypage:

```

1 \LWR@ProvidesPackageDrop{emptypage}

```

File 54 **lwarp-endfloat.sty**

\$ 140 Package **endfloat**

Pkg endfloat endfloat is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{endfloat}

```

2 \newcommand\figureplace{}
3 \newcommand\tableplace{}
4 \newcommand\floatplace[1]{}
5 \newcounter{posttable}
6 \newcounter{postfigure}
7 \newcommand*{\theposttbl}{}
8 \newcommand*{\thepostfig}{}
9 \newcommand{\AtBeginFigures}[1]{}
10 \newcommand{\AtBeginTables}[1]{}
11 \newcommand{\AtBeginDelayedFloats}[1]{}
12 \newcommand*{\processdelayedfloats}{}
13 \newcommand*{\efloatseparator}{}

```

File 55 **lwarp-endheads.sty**

\$ 141 Package **endheads**

Pkg endheads endheads is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{endheads}

```

2 \newcommand{\changesinglepageabbrev}[1]{}
3 \newcommand{\changemultiplepageabbrev}[1]{}
4 \newcommand{\changenotesname}[1]{}
5 \newcommand{\changenotesheader}[1]{}
6 \newcommand{\changenotescontentsname}[1]{}
7 \newcommand{\changechapternotesline}[1]{}
8 \newcommand{\checknoteheaders}{}
9 \newif\ifnotesincontentson \notesincontentsonfalse
10 \newcommand{\notesincontents}{\notesincontentsontrue}
11 \newif\ifendnoteheaderson \endnoteheadersonfalse
12 \newcommand{\setupendnoteheaders}{%
13     \endnoteheadersontrue%
14 }
15 \newif\iftitleinnotes \titleinnotestru
16 \newcommand{\styleforchapternotebegin}{}
17 \newcommand{\styleforchapternoteend}{}
18 \newcommand{\setstyleforchapternotebegin}[1]{%
19     \renewcommand{\styleforchapternotebegin}{#1}%
20 }
21 \newcommand{\setstyleforchapternoteend}[1]{%
22     \renewcommand{\styleforchapternoteend}{#1}%
23 }
24 \newcommand{\resetendnotes}{}
25 \newif\ifnotesbychapteron \notesbychapteronfalse
26 \newcommand{\notesbychapter}{\notesbychapterontrue}

```

File 56 **lwarp-endnotes.sty**

§ 142 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 57 **lwarp-enumerate.sty**

§ 143 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 58 **lwarp-enumitem.sty**

§ 144 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 59 `lwarp-epigraph.sty`

§ 145 Package **epigraph**

(Emulates or patches code by PETER WILSON.)

Pkg epigraph epigraph is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{epigraph}

2 \DeclareDocumentCommand{\qitem}{m m}
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 \DeclareDocumentCommand{\epigraph}{m m}
14 {
15 \begin{LWR@BlockClassWP}{\LWR@origmbox{text-align:right}}{}{epigraph}
16 \qitem{#1}{#2}
17 \end{LWR@BlockClassWP}
18 }
19
20 \DeclareDocumentEnvironment{epigraphs}{}
21 {\LWR@BlockClassWP{\LWR@origmbox{text-align:right}}{}{epigraph}}
22 {\endLWR@BlockClassWP}

```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```

23 \newenvironment*{flushepinormal}{}{}

24 \@ifclassloaded{memoir}{
25 \setlength{\epigraphwidth}{.5\linewidth}
26 \renewcommand{\textflush}{flushepinormal}
27 \renewcommand{\epigraphhead}[2][0]{#2}
28 \renewcommand{\dropchapter}[1]{}
29 \renewcommand*{\undodrop}{}
30 }{% not memoir
31 \newlength{\epigraphwidth}
32 \setlength{\epigraphwidth}{.5\linewidth}
33 \newcommand{\textflush}{flushepinormal}
34 \newcommand{\epigraphflush}{flushright}
35 \newcommand{\sourceflush}{flushright}
36 \newcommand*{\epigraphsize}{\small}
37 \newlength{\epigraphrule}
38 \newlength{\beforeepigraphskip}
39 \newlength{\afterepigraphskip}
40 \newcommand{\epigraphhead}[2][0]{#2}
41 \newcommand{\dropchapter}[1]{}
42 \newcommand*{\undodrop}{}
43 }% not memoir
44
45 \let\cleartoevenpage\relax% also in nextpage
46 \newcommand{\cleartoevenpage}[1][{}]{

```

File 60 **lwarp-eso-pic.sty**

§ 146 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 \newcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*{\ClearShipoutPictureBG}{}
16 \newcommand*{\ClearShipoutPicture}{}
17 \newcommand*{\ClearShipoutPictureFG}{}
18 \newcommand{\gridSetup}[6] [] {}

```

File 61 **lwarp-everypage.sty**

§ 147 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 62 **lwarp-everyshi.sty**

§ 148 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 63 **lwarp-extramarks.sty**

§ 149 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 64 **lwarp-fancybox.sty**

§ 150 Package **fancybox**

(Emulates or patches code by TIMOTHY VAN ZANDT.)

Pkg fancybox fancybox is supported with some patches.

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

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

```

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

framed table example The fancybox documentation's example framed table using an \fbox containing a tabular does not work with lwarp, but the FramedTable environment does work if \fbox is replaced by \fboxBlock. This method loses HTML formatting. A better

method is to enclose the table's contents inside a `fminipage` environment. The caption may be placed either inside or outside the `fminipage`:

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

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

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

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

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

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 `UseVerbatim` and friends:

```
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*{-\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 65 **lwarp-fancyheadings.sty**

§ 151 Package **fancyheadings**

Pkg fancyheadings fancyheadings is superceded by fancyhdr.

for HTML output: 1 \LWR@loadnever{fancyheadings}{fancyhdr}

File 66 **lwarp-fancyhdr.sty**

§ 152 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 67 **lwarp-fancyref.sty**

§ 153 Package **fancyref**

Pkg fancyref fancyref is emulated.

for HTML output: 1 \LWR@ProvidesPackagePass{fancyref}

To remove the margin option, if \fancyrefhook is anything other than the paren option, then force it to the default instead. (Comparing to the margin option was not possible since lwarp has revised the meaning of \mbox so the comparison failed.)

```

2 \newcommand*{\LWRfref@parenfancyrefhook}[1] {(#1)}
3
4 \ifdefstrequal{\fancyrefhook}{\LWRfref@parenfancyrefhook}
5 {}{
6   \renewcommand*{\fancyrefhook}[1] {#1}%
7 }

```

Modified to ignore the page number and varioref.

```

8 \renewcommand*{\@f@ref}[4] {%
9   \@ifundefined{#1r@#2@#3} {%
10     \PackageError{fancyref} {%
11       \backslashchar#1ref\space format ‘‘#2’’
12       undefined\MessageBreak
13       for label type ‘‘#3’’%
14     } {%
15       The format ‘‘#2’’ was not defined for the label type
16       ‘‘#3’’\MessageBreak
17       and the \backslashchar#1ref\space command. Perhaps
18       you have only misspelled its name.\MessageBreak
19       Otherwise you will have to define it with
20       \protect\new#1refformat\MessageBreak
21       prior to using it.%
22     }%
23   } {%

```

```

24    \fancyrefhook{%
25        \@nameuse{#1r@#2@#3}%
26        {\ref{#3\fancyrefargdelim#4}}%
27    %    {\pageref{#3\fancyrefargdelim#4}}% original
28    %    {\@fancyref@page@ref{#3\fancyrefargdelim#4}}% original
29        {}% lwarp
30        {}% lwarp
31    }%
32 }%
33 }%

```

File 68 **lwarp-fancyvrb.sty**

§ 154 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.}

12 \preto\FVB@Verbatim{\LWR@forcenewpage}
13 \preto\FVB@LVerbatim{\LWR@forcenewpage}
14 % \preto\FVB@BVerbatim{\LWR@forcenewpage}% Fails, so done below.

```

Simplified to remove PDF formatting:

```

15 \def\FV@BeginListFrame@Single{%
16   \FV@SingleFrameLine{\z@}%
17 }
18
19 \def\FV@endListFrame@Single{%
20   \FV@SingleFrameLine{\@ne}%
21 }
22
23 \def\FV@BeginListFrame@Lines{%
24   \FV@SingleFrameLine{\z@}%
25 }
26
27 \def\FV@endListFrame@Lines{%
28   \FV@SingleFrameLine{\@ne}%
29 }
30
31 \renewcommand*\FV@SingleFrameSep{}
```

Adds HTML formatting:

```

32 \def\FV@BUseVerbatim#1{%
33   \LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}%
34   \FV@BVerbatimBegin#1\FV@BVerbatimEnd%
35   \LWR@afterendverbatim%
36 }
```

`\LWR@FVstyle` Holds the style of the verbatim.

```

37 \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.

```

38 \newcommand*\LWR@fvstartnone{%
39   \LWR@traceinfo{fvstartnone}%
40   % \hbox to\z@{
41   \LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}%
42   % }%
43 }
44
45 \newcommand*\LWR@fvendnone{%
46   \LWR@traceinfo{fvendnone}%
47   % \hbox to\z@{
48   \LWR@afterendverbatim%
49   % }%
```

```

50 }
51
52 \newcommand*{\LWR@fvstartsingle}{%
53 \LWR@traceinfo{fvstartsingle}%
54 \LWR@fvstartnone%
55 \FV@BeginListFrame@Single%
56 }
57
58 \newcommand*{\LWR@fvendsingle}{%
59 \LWR@traceinfo{fvendsingle}%
60 \FV@endListFrame@Single%
61 \LWR@fvendnone%
62 }
63
64 \newcommand*{\LWR@fvstartline}{%
65 \LWR@traceinfo{fvstartline}%
66 \LWR@fvstartnone%
67 % \setlength{\LWR@templengthone}{\baselineskip}%
68 \FV@BeginListFrame@Lines%
69 % \setlength{\baselineskip}{\LWR@templengthone}%
70 % \setlength{\baselineskip}{5pt}%
71 }
72
73 \newcommand*{\LWR@fvendline}{%
74 \LWR@traceinfo{fvendline}%
75 \FV@endListFrame@Lines%
76 \LWR@fvendnone%
77 }

```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```

78 \newcommand*{\LWR@FVfindbordercolor}{%
79 \FancyVerbRuleColor%
80 \LWR@findcurrenttextcolor%
81 \color{black}%
82 }
83
84 % border width of \FV@FrameRule
85 \newcommand*{\LWR@FVborderstyle}[1]{%
86 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
87 \LWR@FVfindbordercolor %
88 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ; %
89 }
90
91 \def\FV@Frame@none{%
92 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
93 \let\FV@BeginListFrame\LWR@fvstartnone%
94 \let\FV@LeftListFrame\relax%

```

```

95 \let\FV@RightListFrame\relax%
96 \let\FV@EndListFrame\LWR@fvendnone}
97
98 \FV@Frame@none% default values
99
100 \def\FV@Frame@single{%
101 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{}}}%
102 \let\FV@BeginListFrame\LWR@fvstartsingle%
103 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
104 \let\FV@RightListFrame\FV@RightListFrame@Single%
105 \let\FV@EndListFrame\LWR@fvendsingle}
106
107 \def\FV@Frame@lines{%
108 \renewcommand*{\LWR@FVstyle}{%
109   \LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}}}%
110 }%
111 \let\FV@BeginListFrame\LWR@fvstartline%
112 \let\FV@LeftListFrame\relax%
113 \let\FV@RightListFrame\relax%
114 \let\FV@EndListFrame\LWR@fvendline}
115
116 \def\FV@Frame@topline{%
117 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}}%
118 \let\FV@BeginListFrame\LWR@fvstartline%
119 \let\FV@LeftListFrame\relax%
120 \let\FV@RightListFrame\relax%
121 \let\FV@EndListFrame\LWR@fvendnone}
122
123 \def\FV@Frame@bottomline{%
124 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}}%
125 \let\FV@BeginListFrame\LWR@fvstartnone%
126 \let\FV@LeftListFrame\relax%
127 \let\FV@RightListFrame\relax%
128 \let\FV@EndListFrame\LWR@fvendline}
129
130 \def\FV@Frame@leftline{%
131 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}}%
132 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
133 \ifx\FancyVerbFillColor\relax%
134 \let\FV@FrameFillLine\relax%
135 \else%
136 \@tempdima\FV@FrameRule\relax%
137 \multiply\@tempdima-\tw@%
138 \edef\FV@FrameFillLine{%
139 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
140 \kern-\number\@tempdima sp}}}%
141 \fi%
142 \let\FV@BeginListFrame\LWR@fvstartnone%
143 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
144 \let\FV@RightListFrame\relax%

```

```
145 \let\FV@EndListFrame\LWR@fvendnone}
```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```
146 \def\FV@SingleFrameLine#1{%
147 %   \hbox to\z@{%
148 %       \kern\leftmargin
149 %       \ifnum#1=\z@\relax
150 %       \let\FV@Label\FV@LabelBegin
151 %       \else
152 %       \let\FV@Label\FV@LabelEnd
153 %       \fi
154 %       \ifx\FV@Label\relax
155 %       \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
156 %       \else
157 %       \ifnum#1=\z@
158 %       \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
159 %       \ifx\FV@LabelPositionTopLine\relax
160 %       \else
161 %       \LWR@FVfindbordercolor
162 %       \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
163 %       \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
164 %       \LWR@htmltagc{/div}
165 %       \fi
166 %       \else
167 %       \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
168 %       \ifx\FV@LabelPositionBottomLine\relax
169 %       \else
170 %       \LWR@FVfindbordercolor
171 %
172 %       \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
173 %       \LWR@origtextrm{\FV@LabelEnd}
174 %       \LWR@htmltagc{/div}
175 %       \fi
176 %       \fi
177 %       \fi
178 %       \hss
179 %       }
180 }
```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```
181 \def\FV@ListProcessLine#1{%
182   \hbox to \hsize{%
183 %     \kern\leftmargin
184 %     \hbox to \VerbatimHTMLWidth {%
185 %     \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%

```

```

186         \FV@LeftListNumber%
187 %         \FV@LeftListFrame
188         \FancyVerbFormatLine{#1}%
189         \hss%
190 %         \FV@RightListFrame
191         \FV@RightListNumber%
192     }%
193     \hss% required to avoid underfull hboxes
194 }
195 }

```

Env BVerbatim

```

196 \AtBeginEnvironment{BVerbatim}
197 {%
198 \LWR@forcenewpage% instead of \preto
199 \LWR@atbeginverbatim{bverbatim}%
200 }
201
202 \AfterEndEnvironment{BVerbatim}
203 {%
204 \leavevmode\par\LWR@origvspace{-\baselineskip}%
205 \LWR@afterendverbatim%
206 }

```

End of the modifications to make at the end of the preamble:

```

207 } % \AfterEndPreamble

208 \end{warpHTML}

```

File 69 **lwarp-figcaps.sty**

§ 155 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 70 **lwarp-figsize.sty**

§ 156 Package **figsize**

(Emulates or patches code by ANTHONY A. TANBAKUCHI.)

Pkg **figsize** figsize is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{figsize}

Emulates a virtual 6×9 inch textsize.

```

2 \newlength{\figwidth}
3 \newlength{\figheight}
4
5 \newcommand{\SetFigLayout}[3][0]{%
6 \setlength{\figheight}{8in}%
7 \setlength{\figheight}{\figheight / #2}%
8 %
9 \setlength{\figwidth}{5.5in}%
10 \setlength{\figwidth}{\figwidth / #3}%
11 }


```

File 71 **lwarp-fix2col.sty**

§ 157 Package **fix2col**

Pkg **fix2col** fix2col is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fix2col}

File 72 **lwarp-fixme.sty**§ 158 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 73 **lwarp-fixmetodonotes.sty**

§ 159 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 74 **lwarp-flafter.sty**

§ 160 Package **flafter**

Pkg flafter flafter is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{flafter}
 2 \providecommand\fl@trace[1]{}

File 75 **lwarp-float.sty**

§ 161 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 66.2 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 76 **lwarp-floatflt.sty**

§ 162 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 $\{\langle type \rangle\} \{\langle width \rangle\}$ Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m m}{
3 {%
4 \setlength{\LWR@templengthone}{#3}%
5 \LWR@BlockClassWP{%
6    float:right; %
7    width:\LWR@printlength{\LWR@templengthone}; %
8    margin:10pt%
```

```

9 }{%
10   width:\LWR@printlength{\LWR@templengthone}%
11 }%
12 {marginblock}%
13 \captionsetup{type=#2}%
14 }
15 {%
16 \endLWR@BlockClassWP%
17 }

```

```

Env floatingfigure  [<placement>] [<width>]

18 \DeclareDocumentEnvironment{floatingfigure}{o m}
19   {\begin{KFLTfloatflt@marginfloat}{figure}{#2}}
20   {\end{KFLTfloatflt@marginfloat}}

Env floatingtable  [<placement>]

21 \DeclareDocumentEnvironment{floatingtable}{o}
22   {\begin{KFLTfloatflt@marginfloat}{table}{1.5in}}
23   {\end{KFLTfloatflt@marginfloat}}

```

File 77 **lwarp-floatpag.sty**

§ 163 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 78 **lwarp-floatrow.sty**

§ 164 Package **floatrow**

(Emulates or patches code by OLGA LAPKO.)

Pkg floatrow floatrow is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{floatrow}

⚠ misplaced alignment alignment tab character & Use \StartDefiningTabulars and \EndDefiningTabulars before and after defining macros using \ttabbox with a tabular inside. See section 8.8.

⚠ 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` $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```

67 \newfloatcommand{\ffigbox}{figure}[\nocapbeside] []

```

`\ttabbox` $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```

68 \newfloatcommand{\ttabbox}{table}[\captop] [\FBwidth]

```

`\fcapside` $[\langle width \rangle] [\langle height \rangle] [\langle vposn \rangle] \{\langle caption commands \rangle\} \{\langle contents \rangle\}$

```

69 \newfloatcommand{\fcapside}{figure}[\capbeside] []

```

Env `floatrow` $[\langle numfloats \rangle]$

The row of floats is placed into a <div> of class floatrow.

```
70 \newenvironment*{floatrow}[1][2]
71 {
72 \LWR@forcenewpage
73 \BlockClass{floatrow}
```

While inside the floatrow, divide the \linewidth by the number of floats.

```
74 \booltrue{LWR@infloatrow}
75 \setlength{\linewidth}{6in/#1}
76 }
77 {
78 \boolfalse{LWR@infloatrow}
79 \endBlockClass
80 }
```

Keys for \DeclareNewFloatType:

```
81 \newcommand*{\LWR@frowkeyplacement}{}
82 \newcommand*{\LWR@frowkeyname}{}
83 \newcommand*{\LWR@frowkeyfileext}{}
84 \newcommand*{\LWR@frowkeywithin}{}
85 \newcommand*{\LWR@frowkeycapstyle}{}
86
87 \define@key{frowkeys}{placement}{}%
88 \define@key{frowkeys}{name}{\renewcommand{\LWR@frowkeyname}{#1}}%
89 \define@key{frowkeys}{fileext}{\renewcommand{\LWR@frowkeyfileext}{#1}}%
90 \define@key{frowkeys}{within}{\renewcommand{\LWR@frowkeywithin}{#1}}%
91 \define@key{frowkeys}{relatedcapstyle}{}%
```

\DeclareNewFloatType {<type>} {<options>}

Use \listof{type}{Title} to print a list of the floats.

```
92 \newcommand*{\DeclareNewFloatType}[2]{%
```

Reset key values:

```
93 \renewcommand*{\LWR@frowkeyplacement}{}%
94 \renewcommand*{\LWR@frowkeyname}{}%
95 \renewcommand*{\LWR@frowkeyfileext}{}%
96 \renewcommand*{\LWR@frowkeywithin}{}%
97 \renewcommand*{\LWR@frowkeycapstyle}{}%
```

Read new key values:

```
98 \LWR@traceinfo{about to setkeys frowkeys}%
99 \setkeys{frowkeys}{#2}%
100 \LWR@traceinfo{finished setkeys frowkeys}%
```

Create a new float with optional [within]:

```
101 \ifthenelse{\equal{\LWR@frowkeywithin}{} }%
```

```

102 {%
103   \LWR@traceinfo{about to newfloat #1 \LWR@frowkeyplacement\ %
104     \LWR@frowkeyfileext}%
105   \newfloat{#1}{\LWR@frowkeyplacement}{\LWR@frowkeyfileext}%
106 }%
107 {%
108   \LWR@traceinfo{about to newfloat #1\ \LWR@frowkeyplacement\ %
109     \LWR@frowkeyfileext\ \LWR@frowkeywithin}%
110   \newfloat{#1}{\LWR@frowkeyplacement}%
111   {\LWR@frowkeyfileext}{\LWR@frowkeywithin}%
112   \LWR@traceinfo{finished newfloat #1}%
113 }%

```

Rename the float if a name was given:

```

114 \ifthenelse{\equal{\LWR@frowkeyname}{}}{%
115 }{%
116 {\floatname{#1}{\LWR@frowkeyname}}%
117 }

```

Not used:

```

118 \newcommand{\buildFBBBOX}[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 [<num_floats>]
```

```
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 79 **lwarp-fltrace.sty**

\$ 165 Package **fltrace**

Pkg fltrace fltrace is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fltrace}

2 \def\tracefloats{}
3 \def\tracefloatsoff{}
4 \def\tracefloatvals{}

```

File 80 **lwarp-flushend.sty**

\$ 166 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 81 **lwarp-fncychap.sty**

§ 167 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 82 **lwarp-fnlineno.sty**

§ 168 Package **fnlineno**

Pkg fnlineno fnlineno is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fnlineno}

File 83 **lwarp-fnpos.sty**

§ 169 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 84 **lwarp-fontenc.sty**

§ 170 Package **fontenc**

Pkg fontenc If using pdf_{La}TeX, 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 85 **lwarp-fontspec.sty**

§ 171 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 86 **lwarp-footmisc.sty**

§ 172 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 87 **lwarp-footnote.sty**

§ 173 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 \def\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 88 **lwarp-footnotehyper.sty**

§ 174 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 89 **lwarp-footnpag.sty**

§ 175 Package **footnpag**

Pkg footnpag footnpag is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{footnpag}

File 90 **lwarp-framed.sty**

§ 176 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]{shaded}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
```

```

25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \#\LWR@tempcolor]{shaded}%
28 }
29 {\endBlockClass}
30
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage
34   \BlockClass{framedleftbar}
35   \def\FrameCommand{}%
36   \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47
48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
52 }
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{0pt}{0pt}{0pt}{0pt}{\BODY}
61 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{\% not empty
67   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68   \textcolor{TFTitleColor}{\textbf{#1}}%
69   \end{BlockClass}

```

```

70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{}{% not empty
75     \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76     \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77     \textcolor{TFTitleColor}{\textbf{#2}}%
78     \end{BlockClass}
79 }% not empty
80 \end{BlockClass}
81 }

\TitleBarFrame [<marker>] {<title>} {<contents>}

82 \renewcommand\TitleBarFrame[3][]{
83 \CustomFBox
84     {#2}{}%
85     \fboxrule\fboxrule\fboxrule\fboxrule
86     {#3}%
87 }

88 \renewcommand{\TF@Title}[1]{#1}

MakeFramed {<settings>}

89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
94 }

\fb@put@frame {<frame cmd no split>} {<frame cmd split>}

95 \renewcommand*\fb@put@frame[2]{%
96 \relax%
97 \@tempboxa%
98 }

```

File 91 **lwarp-ftnright.sty**

§ 177 Package **ftnright**

Pkg ftnright ftnright is ignored.

for HTML output: Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

File 92 **lwarp-fullpage.sty**

§ 178 Package **fullpage**

Pkg fullpage fullpage is ignored.

for HTML output: Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

File 93 **lwarp-fullwidth.sty**

§ 179 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 94 **lwarp-fwlw.sty**

§ 180 Package **fwlw**

Pkg fwlw fwlw is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fwlw}

```

2 \newbox\FirstWordBox      \global\setbox\FirstWordBox\hbox{}
3 \newbox\NextWordBox       \global\setbox\NextWordBox\hbox{}
4 \newbox\LastWordBox       \global\setbox\LastWordBox\hbox{}
5 \def\ps@fwlhead{}
6 \def\ps@NextWordFoot{}

```

File 95 **lwarp-geometry.sty**

§ 181 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 96 **lwarp-glossaries.sty**

§ 182 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 97 **lwarp-graphics.sty**

§ 183 Package **graphics**

(Emulates or patches code by D. P. CARLISLE.)

Pkg graphics graphics is emulated.

for HTML output: 1 \LWR@ProvidesPackagePass{graphics}

§ 183.1 **Graphics extensions**

\DeclareGraphicsExtensions $\{\langle list \rangle\}$

\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 }
```

§ 183.2 **Length conversions and graphics options**



whitespace

A scaled image in \LaTeX by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

Used to store the user's selected dimensions and HTML class.

The class defaults to “inlineimage” unless changed by a `class=xyx` option.

```
10 \newlength{\LWR@igwidth}
11 \newlength{\LWR@igheight}
12 \newcommand*{\LWR@igwidthstyle}{}
13 \newcommand*{\LWR@igheightstyle}{}
14 \newcommand*{\LWR@igorigin}{}
15 \newcommand*{\LWR@igangle}{}

```

```

16 \newcommand*\LWR@igxscale{1}
17 \newcommand*\LWR@igyscale{1}
18 \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:

```

19 \define@key{igraph}{width}{%
20 \setlength{\LWR@igwidth}{#1}%
21 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}%
22 {%

```

Default to use the converted fixed length given:

```

23 \renewcommand*\LWR@igwidthstyle{width:\LWR@printlength{\LWR@igwidth}}%

```

If ex or em dimensions were given, use those instead:

```

24 \IfEndWith{#1}{ex}%
25 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes ex
26 {}% not ex
27 \IfEndWith{#1}{em}%
28 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes em
29 {}% not em
30 \IfEndWith{#1}{\}%
31 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes percent
32 {}% not percent
33 \IfEndWith{#1}{px}%
34 {\renewcommand*\LWR@igwidthstyle{width:#1}}% yes px
35 {}% not px
36 }{}% end of length > 0pt
37 }

```

If an optional height was given, set an HTML style:

```

38 \define@key{igraph}{height}{%
39 \setlength{\LWR@igheight}{#1}%
40 \ifthenelse{\lengthtest{\LWR@igheight > 0pt}}%
41 {%

```

Default to use the converted fixed length given:

```

42 \renewcommand*\LWR@igheightstyle{%
43 height:\LWR@printlength{\LWR@igheight} %
44 }%

```

If ex or em dimensions were given, use those instead:

```

45 \IfEndWith{#1}{ex}%
46 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
47 {}% not ex
48 \IfEndWith{#1}{em}%
49 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
50 {}% not em
51 \IfEndWith{#1}{\}%
52 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
53 {}% not percent
54 \IfEndWith{#1}{px}%
55 {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
56 {}% not px
57 }{}% end of length > Opt
58 }

```

Handle origin key:

```

59 \define@key{igraph}{origin}{%
60 \renewcommand*{\LWR@igorigin}{#1}%
61 }

```

Handle angle key:

```

62 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```

63 \define@key{igraph}{class}{\renewcommand*{\LWR@igclass}{#1}}
64

```

It appears that graphicx does not have separate keys for xscale and yscale. scale adjusts both at the same time.

```

65 \define@key{igraph}{scale}{%
66 \renewcommand*{\LWR@igxscale}{#1}%
67 \renewcommand*{\LWR@igyyscale}{#1}}

```

Numerous ignored keys:

```

68 \define@key{igraph}{bb}{}
69 \define@key{igraph}{bblx}{}
70 \define@key{igraph}{bbly}{}
71 \define@key{igraph}{bburx}{}
72 \define@key{igraph}{bbury}{}
73 \define@key{igraph}{natwidth}{}
74 \define@key{igraph}{natheight}{}

```

```

75 \define@key{igraph}{hiresbb}{}
76 \define@key{igraph}{viewport}{}
77 \define@key{igraph}{trim}{}
78 \define@key{igraph}{totalheight}{}
79 \define@key{igraph}{keepaspectratio}{}
80 \define@key{igraph}{clip}{}
81 \define@key{igraph}{draft}{}
82 \define@key{igraph}{type}{}
83 \define@key{igraph}{ext}{}
84 \define@key{igraph}{read}{}
85 \define@key{igraph}{command}{}

```

§ 183.3 **Printing HTML styles**

`\LWR@rotstyle` $\{\langle prefix \rangle\} \{\langle degrees \rangle\}$

Prints the rotate style with the given prefix.

prefix is -ms- or -webkit- or nothing, and is used to generate three versions of the transform:rotate style.

```

86 \newcommand*{\LWR@rotstyle}[2]{%
87   #1transform:rotate(-#2deg);
88 }

```

`\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.

```

89 \newcommand*{\LWR@scalestyle}[3]{%
90   #1transform:scale(#2,#3);
91 }

```

§ 183.4 **\includegraphics**

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```

92 \newbool{LWR@infloatrow}
93 \boolfalse{LWR@infloatrow}

```

`\LWR@opacity` may be set by the transparent package. For HTML it is only used for `\includegraphics`.

```

94 \def\LWR@opacity{1}

```

Used to determine the actual image size if needed:

```
95 \newsavebox{\LWR@imagesizebox}

96 \let\LWR@origGin@setfile\Gin@setfile
```

Define the new class key for the print-mode version of `\includegraphics`, which is enabled inside a `lateximage`.

```
97 \AtBeginDocument{
98 \define@key{Gin}{class}{}
99 }
```

```
\LWR@includegraphicsb * [2: options] [3: options] {4: filename}
```

graphics syntax is `\includegraphics * [llx, lly] [urx, ury] {file}`

graphicx syntax is `\includegraphics [key values] {file}`

If #3 is empty, only one optional argument was given, thus `graphicx` syntax.

```
100 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
101 {%
102 \LWR@traceinfo{\LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
103 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
104 \begingroup%
105 \ifthenelse{\cinttest{\value{\LWR@minipagedepth}}{=}{0}}{%
106 {%
107   \ifbool{\LWR@infloatrow}%
108   {}
109   {% not in a minipage or a floatrow:
110     \setlength{\linewidth}{6in}%
111     \setlength{\textwidth}{6in}%
112     \setlength{\textheight}{9in}%
113   }%
114 }{}}%

115 \begingroup%
116 \renewcommand*{\Gin@setfile}[3]{%
117 \LWR@traceinfo{\Gin@setfile ##3}%
```

```

118 \xdef\LWR@parsedfilename{##3}%
119 }%
120 \Gininclude@graphics{\detokenize\expandafter{#4}}%
121 \endgroup%
122 \filename@parse{\LWR@parsedfilename}%
123 \LWR@traceinfo{\LWR@parsedfilename is \LWR@parsedfilename}%
124 % \LWR@sanitize{\LWR@parsedfilename}%

```

For correct em sizing during the width and height conversions:

```

125 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

126 \setlength{\LWR@igwidth}{0pt}%
127 \setlength{\LWR@igheight}{0pt}%
128 \renewcommand*{\LWR@igwidthstyle}{}%
129 \renewcommand*{\LWR@igheightstyle}{}%
130 \renewcommand*{\LWR@igorigin}{}%
131 \renewcommand*{\LWR@igangle}{}%
132 \renewcommand*{\LWR@igxscale}{1}%
133 \renewcommand*{\LWR@igyyscale}{1}%
134 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```

135 \IfValueF{#3}{%
136 \IfValueTF{#2}%
137 {\setkeys{igraph}{#2}}%
138 {\setkeys{igraph}{}}%
139 }%

```

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:

```

140 \ifbool{FormatWP}{%
141   \begingroup%
142   \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
143   \define@key{Gin}{angle}{}%
144   \IfBooleanTF{#1}%
145   {% starred
146     \IfValueTF{#3}%
147     {%
148       \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] [#3] {#4}}%
149     }%
150     {%
151       \IfValueTF{#2}%
152       {%
153         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] {#4}}%

```

```

154      }{%
155      \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics*{#4}}%
156      }%
157    }%
158  }% starred
159  {% not starred
160    \IfValueTF{#3}%
161    {%
162      \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics[#2][#3]{#4}}%
163    }%
164    {%
165      \IfValueTF{#2}%
166      {%
167        \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics[#2]{#4}}%
168      }{%
169        \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics{#4}}%
170      }%
171    }%
172  }% not starred
173  \endgroup%
174  \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
175  \global\renewcommand*{\LWR@igwidthstyle}{width:\LWR@printlength{\LWR@igwidth}}%
176  \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
177  \global\renewcommand*{\LWR@igheightstyle}{height:\LWR@printlength{\LWR@igheight}}%
178 }{}}%

```

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.

```

179 \LWR@traceinfo{\LWR@includegraphicsb: about to create href}%
180 \href{\LWR@parsedfilename}%
181 {% start of href
182 \LWR@traceinfo{\LWR@includegraphicsb: about to LWR@htmltag}%
183 \LWR@htmltag{% start of image tags
184 img src="%
185 \begingroup\@sanitize\LWR@parsedfilename\endgroup%
186 " \LWR@orignewline%

```

Only include a style tag if a width, height, angle, or scale was given:

```

187 \ifthenelse{
188   \NOT\equal{\LWR@igwidthstyle}{ } \OR
189   \NOT\equal{\LWR@igheightstyle}{ } \OR
190   \NOT\equal{\LWR@igorigin}{ } \OR
191   \NOT\equal{\LWR@igangle}{ } \OR
192   \NOT\equal{\LWR@igxscale}{1} \OR

```

```

193     \NOT\equal{\LWR@igyscale}{1}
194 }%
195 {\LWR@origtilde{} style="%
196 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{}}%
197 {\LWR@igwidthstyle;}{}}%
198 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{}}%
199 {\LWR@igheightstyle;}{}}%
200 \ifthenelse{\NOT\equal{\LWR@igorigin}{}}%
201 {\LWR@origtilde{} transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{}}%
202 \ifthenelse{\NOT\equal{\LWR@igangle}{}}%
203 {%
204 \LWR@rotstyle{-ms-}{\LWR@igangle} %
205 \LWR@rotstyle{-webkit-}{\LWR@igangle} %
206 \LWR@rotstyle{}{\LWR@igangle} %
207 }%
208 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}}{OR%
209 \NOT\equal{\LWR@igyscale}{1}}%
210 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale} %
211 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale} %
212 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}} %
213 %
214 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
215 {opacity:\LWR@opacity; }%
216 {}%
217 %
218 " \LWR@orignewline}{}}%

```

Set the class:

```

219 \LWR@origtilde{} class="\LWR@igclass" \LWR@orignewline%
220 }% end of image tags
221 }% end of href

```

Return to original page size and font size:

```

222 \endgroup
223 \LWR@traceinfo{\LWR@includegraphicsb done}%
224 }

```

`\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.

```

225 \AtBeginDocument{
226

```

```

227 \LWR@traceinfo{Patching includegraphics.}
228
229 \LetLtxMacro\LWR@originincludegraphics\includegraphics
230
231 \renewcommand*\includegraphics{
232 {%

```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```

233 \LWR@traceinfo{includegraphics}%
234 \LWR@ensuredoingapar%
235 \LWR@includegraphicsb%
236 }% includegraphics
237 }% AtBeginDocument

```

§ 183.5 Boxes

`\LWR@rotboxorigin` Holds the origin key letters.

```

238 \newcommand*\LWR@rotboxorigin{}

```

`\LWR@originname` $\{ \langle letter \rangle \}$

Given one \LaTeX origin key value, translate into an HTML origin word:

```

239 \newcommand*\LWR@originname[1]{%
240 \ifthenelse{\equal{#1}{t}}{top}{}%
241 \ifthenelse{\equal{#1}{b}}{bottom}{}%
242 \ifthenelse{\equal{#1}{c}}{center}{}%
243 \ifthenelse{\equal{#1}{l}}{left}{}%
244 \ifthenelse{\equal{#1}{r}}{right}{}%
245 }

```

`\LWR@originnames` $\{ \langle letters \rangle \}$

Given one- or two-letter \LaTeX origin key values, translate into HTML origin words:

```

246 \newcommand*\LWR@originnames[1]{%
247 \StrChar{#1}{1}[\LWR@strresult]%
248 \LWR@originname{\LWR@strresult}
249 \StrChar{#1}{2}[\LWR@strresult]%
250 \LWR@originname{\LWR@strresult}
251 }

```

Handle the origin key for `\rotatebox`:

```

252 \define@key{krotbox}{origin}{%
253 \renewcommand*{\LWR@rotboxorigin}{#1}%
254 }

```

These keys are ignored:

```

255 \define@key{krotbox}{x}{}
256 \define@key{krotbox}{y}{}
257 \define@key{krotbox}{units}{}

```

`\rotatebox` [*keyval list*] {*angle*} {*text*}

```

258 \LetLtxMacro\LWR@origrotatebox\rotatebox
259
260 \AtBeginDocument{
261 \RenewDocumentCommand{\rotatebox}{O{} m +m}{%

```

Reset the origin to “none-given”:

```

262 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```

263 \setkeys{krotbox}{#1}%

```

Select inline-block so that HTML will transform this span:

```

264 \LWR@htmltagc{span style="display: inline-block; %

```

If an origin was given, translate and print the origin information:

```

265 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{} }{%
266 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```

267 \LWR@rotstyle{-ms-}{#2} %
268 \LWR@rotstyle{-webkit-}{#2} %
269 \LWR@rotstyle{}{#2} %
270 " }\LWR@orignewline%

```

Print the text to be rotated:

```

271 \begin{\LWR@nestspan}%
272 #3%

```

Close the span:

```

273 \LWR@htmltagc{/span}%
274 \end{LWR@nestspan}%
275 }
276 }% AtBeginDocument

```

`\scalebox` $\{\langle h-scale \rangle\} [\langle v-scale \rangle] \{\langle text \rangle\}$

```

277 \LetLtxMacro\LWR@origscalebox\scalebox
278
279 \AtBeginDocument{
280 \RenewDocumentCommand{\scalebox}{m o m}{%

```

Select inline-block so that HTML will transform this span:

```

281 \LWR@htmltagc{span style="display: inline-block; %

```

Print the scaling information:

```

282 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
283 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
284 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
285 "%

```

Print the text to be scaled:

```

286 \begin{LWR@nestspan}%
287 #3%

```

Close the span:

```

288 \LWR@htmltagc{/span}%
289 \end{LWR@nestspan}%
290 }
291 }% AtBeginDocument

```

`\reflectbox` $\{\langle text \rangle\}$

```

292 \let\LWR@origreflectbox\reflectbox
293
294 \AtBeginDocument{
295 \renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
296 }

```

`\resizebox` $\{\langle h-length \rangle\} \{\langle v-length \rangle\} \{\langle text \rangle\}$

Simply prints its text argument.

```

297 \LetLtxMacro\LWR@origresizebox\resizebox
298
299 \AtBeginDocument{
300 \renewcommand{\resizebox}[3]{#3}
301 }

```

File 98 **lwarp-graphicx.sty**

§ 184 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 99 **lwarp-grffile.sty**

§ 185 Package **grffile**

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

 **matching PDF and SVG**

lwarp-grffile now exists as a placeholder since **grffile** used to be emulated by **lwarp**, and thus older versions of **lwarp-grffile** may exist and should be overwritten by this newer version.

for HTML output: `1 \LWR@ProvidesPackagePass{grffile}`

File 100 **lwarp-hang.sty**

§ 186 Package **hang**

(Emulates or patches code by ANDREAS NOLDA.)

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   \BlockClass[%
16     \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
17     \LWR@origmbox{text-indent:-\LWR@printlength{\hangingindent}}%
18   ]%
19   {hangingpar}%
20 }
21 {\endBlockClass}
22
23 \newenvironment{hanginglist}
24 {%
25   \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}%
26   \renewcommand*{\LWR@printopenlist}{%
27     \LWR@findhangingleftmargin%
28     ul style="\LWR@origmbox{list-style-type:none;} %
29     \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
30     \LWR@origmbox{text-indent:-\LWR@printlength{\hangingindent}}"%
31   }%
32   \let\item\LWR@itemizeitem%
33   \list{}{}%
34 }
35 {\endlist}
36
37 \newenvironment{compacthang}
38 {\hanginglist}
39 {\endhanginglist}
40
41 \newlength{\labeledleftmargin}
42 \setlength{\labeledleftmargin}{0em}
43
44 \newenvironment{labeledpar}[2]
45 {%
46   \BlockClass[%
47     \LWR@findhangingleftmargin%
48     \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
49     \LWR@origmbox{text-indent:-\LWR@printlength{\hangingindent}}%

```

```

50   ]{\labeledpar}#2%
51 }
52 {\endBlockClass}
53
54 \newenvironment{labeledlist}[1]
55 {\\hanginglist}
56 {\endhanginglist}
57
58 \newenvironment{compactlabel}[1]
59 {\\hanginglist}
60 {\endhanginglist}

```

File 101 **lwarp-hanging.sty**

§ 187 Package **hanging**

Pkg hanging hanging is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{hanging}

```

2 \@ifclassloaded{memoir}{
3 \let\hangpara\relax
4 \let\hangparas\relax
5 \let\endhangparas\relax
6 \let\hangpunct\relax
7 \let\endhangpunct\relax
8 }{}

```

\hangpara {\<indent>} {\<afternum>}

Use hangparas instead.

```

9 \newcommand*{\hangpara}[2]{}

```

Env hangparas {\<indent>} {\<afternum>}

```

10 \newenvironment*{hangparas}[2]
11 {%
12   \BlockClass[%
13     \LWR@origmbox{margin-left:\LWR@printlength{#1}} ; %
14     \LWR@origmbox{text-indent:-\LWR@printlength{#1}}%
15   ]%
16   {\hangingpar}%
17 }
18 {\endBlockClass}

```

Env **hangpunct**

```
19 \newenvironment*{hangpunct}
20 {\BlockClass{hangpunct}}
21 {\endBlockClass}

22 \newcommand{\nhpt}{.}
23 \newcommand{\nhlq}{'}
24 \newcommand{\nhrq}{'}
```

File 102 **lwarp-hypcap.sty**

\$ 188 Package **hypcap**

Pkg hypcap hypcap is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{hypcap}

```
2 \newcommand*{\capstart}{}
3 \newcommand*{\hypcapSPACE}{}
4 \newcommand*{\hypcapredef}[1]{}
5 \newcommand*{\capstartfalse}{}
6 \newcommand*{\capstarttrue}{}

```

File 103 **lwarp-hypdestopt.sty**

\$ 189 Package **hypdestopt**

Pkg hypdestopt hypdestopt is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{hypdestopt}

File 104 **lwarp-hypernat.sty**

\$ 190 Package **hypernat**

Pkg hypernat hypernat is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{hypernat}

File 105 **lwarp-hyperref.sty**

§ 191 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{\hyperimage}{m +m}{%
11 \LWR@ensuredoingapar%
12 \def\LWR@templink{#1}%
13 \@onelevel@sanitize\LWR@templink%
14 \LWR@htmltag{img src="\LWR@templink" alt="#2" class="hyperimage"}%
15 \LWR@ensuredoingapar%
16 }

```

`\hyperdef` `{\langle 1: category \rangle} {\langle 2: name \rangle} {\langle 3: text \rangle}`

Creates an HTML anchor to `category.name` with the given text.

```

17 \NewDocumentCommand{\hyperdef}{m m +m}{%
18 \LWR@ensuredoingapar%
19 \LWR@sublabel{#1.#2}%
20 #3%
21 }

```

`\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.

```

22 \NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%

```

```

23 \def\LWR@templink{#1}%
24 \@onelevel@sanitize\LWR@templink%
25 \def\LWR@templinktwo{#2}%
26 \@onelevel@sanitize\LWR@templinktwo%
27 \def\LWR@templinkthree{#3}%
28 \@onelevel@sanitize\LWR@templinkthree%
29 \LWR@htmltag{a href="\LWR@templink\LWR@hashmark%
30   \LWR@templinktwo.\LWR@templinkthree"%
31 }%
32 #4%
33 \LWR@htmltag{/a}%
34 }

```

`\LWR@hyperrefc` [*<label>*] {<text>}

Creates text as an HTML link to the \LaTeX label.

```

35 \NewDocumentCommand{\LWR@hyperrefc}{O{label} +m}{
36 \LWR@startref{#1}%
37 #2%
38 \LWR@htmltag{/a}%
39 }

```

`\hyperref` {<1: URL>} {<2: category>} {<3: name>} {<4: text>} — or —
[<1: label>] {<2: text>}

```

40 \DeclareRobustCommand*\hyperref{%
41 \LWR@ensuredoingapar%
42 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
43 }

```

`\hypertarget` {<name>} {<text>}

Creates an anchor to name with the given text.

```

44 \NewDocumentCommand{\hypertarget}{m +m}{%
45 \label{#1}%
46 #2%
47 }

```

`\hyperlink` {<name>} {<text>}

Creates a link to the anchor created by `hypertarget`, with the given link text.

Declared because also defined by memoir.

```

48 \DeclareDocumentCommand{\hyperlink}{m +m}{%
49 \hyperref[#1]{#2}%
50 }

```

`\autoref` * $\{\langle label \rangle\}$

For HTML, `\cleveref` is used instead.

```
51 \NewDocumentCommand{\autoref}{s m}{%
52 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
53 }
```

`\autopageref` $\{\langle label \rangle\}$

For HTML, `\cleveref` is used instead.

```
54 \NewDocumentCommand{\autopageref}{s m}{%
55 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
56 }
```

`\pdfstringdef` $\{\langle macroname \rangle\}$ $\{\langle T_{\text{E}}Xstring \rangle\}$

```
57 \newcommand{\pdfstringdef}[2]{}%
```

`\pdfbookmark` $[\langle level \rangle]$ $\{\langle text \rangle\}$ $\{\langle name \rangle\}$

```
58 \newcommand{\pdfbookmark}[3][]{}
```

`\currentpdfbookmark` $\{\langle text \rangle\}$ $\{\langle name \rangle\}$

```
59 \newcommand{\currentpdfbookmark}[2]{}%
```

`\subpdfbookmark` $\{\langle text \rangle\}$ $\{\langle name \rangle\}$

```
60 \newcommand{\subpdfbookmark}[2]{}%
```

`\belowpdfbookmark` $\{\langle text \rangle\}$ $\{\langle name \rangle\}$

```
61 \newcommand{\belowpdfbookmark}[2]{}%
```

`\texorpdfstring` $\{\langle T_{\text{E}}Xstring \rangle\}$ $\{\langle PDFstring \rangle\}$

```
62 \newcommand{\texorpdfstring}[2]{#1}
```

`\hypercalcbp` $\{\langle dimen \rangle\}$ From hyperref.

```
63 \def\hypercalcbp#1{%
64 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
65 }%
```

`\Acrobatmenu` $\{\langle menuoption \rangle\}$ $\{\langle text \rangle\}$

```
66 \newcommand{\Acrobatmenu}[2] {}
```

```
\TextField [\langle parameters \rangle] {\langle label \rangle}
```

```
67 \DeclareRobustCommand{\TextField}[2] [] {}
```

```
\CheckBox [\langle parameters \rangle] {\langle label \rangle}
```

```
68 \DeclareRobustCommand{\CheckBox}[2] [] {}
```

```
\ChoiceMenu [\langle parameters \rangle] {\langle label \rangle} {\langle choices \rangle}
```

```
69 \DeclareRobustCommand{\ChoiceMenu}[3] [] {}
```

```
\PushButton [\langle parameters \rangle] {\langle label \rangle}
```

```
70 \DeclareRobustCommand{\PushButton}[2] [] {}
```

```
\Submit [\langle parameters \rangle] {\langle label \rangle}
```

```
71 \DeclareRobustCommand{\Submit}[2] [] {}
```

```
\Reset [\langle parameters \rangle] {\langle label \rangle}
```

```
72 \DeclareRobustCommand{\Reset}[2] [] {}
```

```
\Gauge [\langle parameters \rangle] {\langle label \rangle}
```

```
73 \DeclareRobustCommand{\Gauge}[2] [] {}
```

```
\LayoutTextField {\langle label \rangle} {\langle field \rangle}
```

```
74 \newcommand*{\LayoutTextField}[2] {}
```

```
\LayoutChoiceField {\langle label \rangle} {\langle field \rangle}
```

```
75 \newcommand*{\LayoutChoiceField}[2] {}
```

```
\LayoutCheckField {\langle label \rangle} {\langle field \rangle}
```

```
76 \newcommand*{\LayoutCheckField}[2] {}
```

```
\MakeRadioField {\langle width \rangle} {\langle height \rangle}
```

```
77 \newcommand*{\MakeRadioField}[2] {}
```

```

\MakeCheckField   {\langle width\rangle} {\langle height\rangle}
                  78 \newcommand*{\MakeCheckField}[2]{\}

\MakeTextField     {\langle width\rangle} {\langle height\rangle}
                  79 \newcommand*{\MakeTextField}[2]{\}

\MakeChoiceField   {\langle width\rangle} {\langle height\rangle}
                  80 \newcommand*{\MakeChoiceField}[2]{\}

\MakeFieldButton   {\langle text\rangle}
                  81 \newcommand{\MakeFieldButton}[1]{\}

```

File 106 **lwarp-hyperxmp.sty**

§ 192 Package **hyperxmp**

Pkg hyperxmp Emulated.

for HTML output: Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 107 **lwarp-hyphenat.sty**

§ 193 Package **hyphenat**

Pkg hyphenat hyphenat is emulated during HTML output, while the print-mode version is used inside a lateximage.

for HTML output: 1 \LWR@ProvidesPackagePass{hyphenat}

```

2 \LetLtxMacro\LWRHYNAT@origtextnhtt\textnhtt
3 \LetLtxMacro\LWRHYNAT@originhttfamily\nhttfamily
4 \LetLtxMacro\LWRHYNAT@orignohyphens\nohyphens
5 \LetLtxMacro\LWRHYNAT@origbshyp\bshyp
6 \LetLtxMacro\LWRHYNAT@origfshyp\fshyp
7 \LetLtxMacro\LWRHYNAT@origdothyp\dothyp
8 \LetLtxMacro\LWRHYNAT@origcolonhyp\colonhyp

```

```

9 \LetLtxMacro\LWRHYNAT@orighyp\hyp
10
11 \LetLtxMacro\textnhtt\texttt
12 \LetLtxMacro\nhttfamily\ttfamily
13
14 \renewcommand{\nohyphens}[1]{#1}
15 \renewrobustcmd{\bshyp}{%
16   \ifmmode\backslash\else\textbackslash\fi%
17 }
18 \renewrobustcmd{\fshyp}{/}
19 \renewrobustcmd{\dothyp}{.}
20 \renewrobustcmd{\colonhyp}{:}
21 \renewrobustcmd{\hyp}{-}
22
23 \appto\LWR@restoreorigformatting{
24 \LetLtxMacro\textnhtt\LWRHYNAT@origtextnhtt
25 \LetLtxMacro\nhttfamily\LWRHYNAT@orignhttfamily
26 \LetLtxMacro\nohyphens\LWRHYNAT@orignohyphens
27 \LetLtxMacro\bshyp\LWRHYNAT@origbshyp
28 \LetLtxMacro\fshyp\LWRHYNAT@origfshyp
29 \LetLtxMacro\dothyp\LWRHYNAT@origdothyp
30 \LetLtxMacro\colonhyp\LWRHYNAT@origcolonhyp
31 \LetLtxMacro\hyp\LWRHYNAT@orighyp
32 }

```

File 108 **lwarp-idxlayout.sty**

§ 194 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 109 **lwarp-ifoddpagel.sty**

§ 195 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\ifoddpagel or oneside
5
6 \DeclareRobustCommand{\checkoddpagel}{\oddpageltrue\oddpagel or onesideltrue}
7
8 \def\oddpagel@pagel{1}
9
10 \def\@ifoddpagel{%
11     \expandafter\@firstoftwo
12 }
13
14 \def\@ifoddpagel or onesidel{%
15     \expandafter\@firstoftwo
16 }

```

File 110 **lwarp-indentfirst.sty**

§ 196 Package **indentfirst**

Pkg indentfirst indentfirst is ignored.

Discard all options for lwarp-indentfirst:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

File 111 **lwarp-inputenc.sty**

§ 197 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 112 **lwarp-keyfloat.sty**

§ 198 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 \RenewDocumentCommand{\KFLT@onefigureimage}{}
4 {%
5 \LWR@traceinfo{KFLT@onefigureimage}%
6 % \begin{lrbox}{\KFLT@envbox}%
7 \ifthenelse{\NOT\equal{\KFLT@lw}{}}{%
8 {\includegraphics%
9 [scale=\KFLT@s,width=\KFLT@imagewidth]{\KFLT@i}}%
10 {% not linewidth
11 \ifthenelse{\dimtest{\KFLT@w}{>}{0pt}}%
12 {% width is given
```

```

13 \ifthenelse{\dimtest{\KFLT@h}{>}{0pt}}%
14 {% w and h
15 \includegraphics%
16 [scale=\KFLT@s,%
17 width=\KFLT@imagewidth,height=\KFLT@h]{\KFLT@i}%
18 }% w and h
19 {% only w
20 \includegraphics%
21 [scale=\KFLT@s,width=\KFLT@imagewidth]{\KFLT@i}%
22 }% only w
23 }% width is given
24 {% width is not given
25 \ifthenelse{\dimtest{\KFLT@h}{>}{0pt}}%
26 {\includegraphics%
27 [scale=\KFLT@s,height=\KFLT@h]{\KFLT@i}}%
28 {\includegraphics%
29 [scale=\KFLT@s]{\KFLT@i}}%
30 }% width is not given
31 }% not linewidth
32 % \end{lrbox}%
33 % \unskip%
34 % \KFLT@findenvboxwidth%
35 % \begin{turn}{\KFLT@r}%
36 % \KFLT@frame{\usebox{\KFLT@envbox}}%
37 % \unskip%
38 % \end{turn}%
39 \LWR@traceinfo{\KFLT@onefigureimage: done}%
40 }

41 \RenewDocumentEnvironment{\KFLT@boxinner}{%
42 {%
43 \LWR@traceinfo{\KFLT@boxinner}%
44 \LWR@stoppars%
45 }
46 {
47 \LWR@startpars%
48 \LWR@traceinfo{\KFLT@boxinner: done}%
49 }

50 \DeclareDocumentEnvironment{\KFLT@marginfloat}{0{-1.2ex} m}
51 {%
52 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{\marginblock}%
53 \captionsetup{type=#2}%
54 }
55 {%
56 \endLWR@BlockClassWP%
57 }

58 \DeclareDocumentEnvironment{\marginfigure}{o}

```

```

59 {\begin{KFLT@marginfloat}{figure}}
60 {\end{KFLT@marginfloat}}
61
62 \DeclareDocumentEnvironment{margintable}{o}
63 {\begin{KFLT@marginfloat}{table}}
64 {\end{KFLT@marginfloat}}

65 \DeclareDocumentEnvironment{keywrap}{m +m}
66 {%
67 \LWR@ensuredoingapar%
68 \setlength{\LWR@templengthone}{#1}%
69 \begin{LWR@BlockClassWP}{%
70     float:right; width:\LWR@printlength{\LWR@templengthone}; %
71     margin:10pt%
72 }%
73 {%
74     width:\LWR@printlength{\LWR@templengthone}%
75 }%
76 {marginblock}%
77 \setlength{\linewidth}{.95\LWR@templengthone}%
78 #2%
79 \end{LWR@BlockClassWP}%
80 }
81 {%
82 }

83 }% AtBeginDocument

```

File 113 **lwarp-layout.sty**

§ 199 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 114 **lwarp-letterspace.sty**

§ 200 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] [] {}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

File 115 **lwarp-lettrine.sty**

§ 201 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 116 **lwarp-lineno.sty**

§ 202 Package **lineno**

(Emulates or patches code by STEPHAN I. BÖTTCHER.)

Pkg lineno lineno is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{lineno}

2 \newcommand*\resetlinenumber[1][\@ne]{}
3
4 \def\linenumbers{%
5     \@ifnextchar[{\resetlinenumber}%]
6         {\@ifstar{\resetlinenumber}{}}%
7     }
8
9 \newcommand*\nolinenumbers{}
10
11 \@namedef{linenumbers*}{\par\linenumbers*}
12 \@namedef{runninglinenumbers*}{\par\runninglinenumbers*}
13
14 \def\endlinenumbers{\par}
15 \let\endrunninglinenumbers\endlinenumbers
16 \let\endpagewiselinenumbers\endlinenumbers
17 \expandafter\let\csname endlinenumbers*\endcsname\endlinenumbers
18 \expandafter\let\csname endrunninglinenumbers*\endcsname\endlinenumbers
19 \let\endnolinenumbers\endlinenumbers
20

```

```

21 \def\pagewiselinenumbers{\linenumbers\setpagewiselinenumbers}
22
23 \def\runninglinenumbers{\setrunninglinenumbers\linenumbers}
24
25 \def\setpagewiselinenumbers{}
26
27 \def\setrunninglinenumbers{}
28
29 \def\linenomath{}%
30 \@namedef{linenomath*}{}%
31 \def\endlinenomath{}
32 \expandafter\let\csname endlinenomath*\endcsname\endlinenomath
33
34 \let\linelabel\label
35
36 \def\switchlinenumbers{\@ifstar{}{}}
37 \def\setmakelinenumbers#1{\@ifstar{}{}}
38
39 \def\leftlinenumbers{\@ifstar{}{}}
40 \def\rightlinenumbers{\@ifstar{}{}}
41
42 \newcounter{linenumber}
43 \newcount\c@pagewiselinenumber
44 \let\c@runninglinenumber\c@linenumber
45
46 \def\runningpagewiselinenumbers{}
47 \def\realpagewiselinenumbers{}
48
49
50 \NewDocumentCommand\modulolinenumbers{s o}{}
51
52 \chardef\c@linenumbermodulo=5
53 \modulolinenumbers[1]
54
55 \newcommand*\firstlinenumber[1]{}
56
57 \newcommand\internallinenumbers{}
58 \let\endinternallinenumbers\endlinenumbers
59 \@namedef{internallinenumbers*}{\internallinenumbers*}
60 \expandafter\let\csname endinternallinenumbers*\endcsname\endlinenumbers
61
62 \newcommand*{\linenoplaceholder}[1]{% redefine per language
63   (line number reference for \detokenize\expandafter{#1})
64 }
65
66 \newcommand*{\lineref}[2][\linenoplaceholder{#2}]
67 \newcommand*{\linerefp}[2][\linenoplaceholder{#2}]
68 \newcommand*{\linerefr}[2][\linenoplaceholder{#2}]
69
70 \newcommand\quotelinenumbers

```

```

71   {\@ifstar\linenumbers{\@ifnextchar[\linenumbers{\linenumbers*}}}
72
73 \newdimen\linenumbersep
74 \newdimen\linenumberwidth
75 \newdimen\quotelinenumbersep
76
77 \quotelinenumbersep=\linenumbersep
78 \let\quotelinenumberfont\linenumberfont
79
80 \def\linenumberfont{\normalfont\tiny\sffamily}
81
82
83 \linenumberwidth=10pt
84 \linenumbersep=10pt
85
86 \def\thelinenumber{}
87
88 \def\LineNumber{}
89 \def\makeLineNumber{}
90 \def\makeLineNumberLeft{}
91 \def\makeLineNumberRight{}
92 \def\makeLineNumberOdd{}
93 \def\makeLineNumberEven{}
94 \def\makeLineNumberRunning{}
95
96
97 \newenvironment{numquote}      {\quote}{\endquote}
98 \newenvironment{numquotation} {\quotation}{\endquotation}
99 \newenvironment{numquote*}     {\quote}{\endquote}
100 \newenvironment{numquotation*}{\quotation}{\endquotation}
101
102 \newdimen\bframerule
103 \bframerule=\fbboxrule
104
105 \newdimen\bframesep
106 \bframesep=\fbboxsep
107
108 \newenvironment{bframe}
109 {%
110   \LWR@forceminwidth{\bframerule}%
111   \BlockClass[
112     border:\LWR@printlength{\LWR@atleastonept} solid black ; %
113     padding:\LWR@printlength{\bframesep}%
114   ]{bframe}
115 }
116 {\endBlockClass}

```

File 117 **lwarp-lips.sty**

§ 203 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 118 **lwarp-listings.sty**

§ 204 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@origlsthkEveryPar\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@origlsthkEveryPar%
21 \else%

```

Inline, so open a :

```

22 \ifbool{\LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
23 \fi%
24 }

```

`\lst@DeInit` Done at the end of a listing.

```

25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the </pre>, then reenable line numbers:

```

27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%

```

Inline, so create the closing :

```

32 \ifbool{\LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%

```

Final listings deinit:

```
34 \LWR@origlst@DeInit%
35 }
```

```
\lst@MakeCaption {</b>}
```

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```
36 \def\lst@MakeCaption#1{%
37 \LWR@traceinfo{MAKING CAPTION at #1}%
38 \lst@ifdisplaystyle
39 \LWR@traceinfo{making a listings display caption}%
40 \ifx #1t%
41 \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
42 \expandafter\refstepcounter
43 \fi {lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 % \ifx\lst@label\@empty\else
46 % \label{\lst@label}\fi
47 \LWR@traceinfo{Finished assigning the label.}%
48 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
49 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
50 \lst@ifnolol\else
51 \ifx\lst@@caption\@empty
52 \ifx\lst@caption\@empty
53 \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54 \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 % \addcontentsline{lol}{lstlisting}{\lst@name}
57 \fi\fi
58 \fi
59 \else
```

This would have to be modified for lwarp:

```
60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61 \addcontentsline{lol}{lstlisting}%
62 {\protect\numberline{\thelstlisting}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64 \fi
65 \fi
66 \fi
67 \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{lst@caption not empty-}%
69 \lst@ifsubstring #1\lst@captionpos
70 {\begingroup
```

```
71 \LWR@traceinfo{at the selected position}%
```

These space and box commands are not needed for HTML output:

```
72 %           \let\@vskip\vskip
73 %           \def\vskip{\afterassignment\lst@vskip \@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@originormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\\%
114     right&\def\lst@PlaceNumber{\rlap{\LWR@originormalfont
115                                     \kern\VerbatimHTMLWidth \kern\lst@numbersep
116                                     \lst@numberstyle{\thelstnumber}}}%
117     }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

118 \end{warpHTML}

```

File 119 **lwarp-longtable.sty**

§ 205 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`.



lateximage longtable is not supported inside a `lateximage`.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` * [*horizontalalignment*] {*colspec*} Emulates the `longtable` environment.

Per the `caption` package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options `[c]`, `[l]`, and `[r]` are thrown away.

```

2 \newenvironment{longtable*}[2][{}]{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{\LTcaption}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][{}]{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{\LTcaption}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{\LTcaption}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25

```

Provided for compatibility, but ignored:

```

26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendoffline}% throws away options // [dim] and // *
28 \def\endfirsthead{\LWR@tabularendoffline}
29 \def\endfoot{\LWR@tabularendoffline}
30 \def\endlastfoot{\LWR@tabularendoffline}
31 \newcommand\tabularnewline{\LWR@tabularendoffline}
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@tabularendoffline}

```

File 120 **lwarp-lscape.sty**

§ 206 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 121 **lwarp-ltcaption.sty**

§ 207 Package **ltcaption**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg **ltcaption** **ltcaption** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcaptype is already defined by lwarp.

longtable* is already defined by lwarp-longtable.

```
2 \newlength{\LTcapskip}
```

```
3 \newlength{\LTcapleft}
```

```
4 \newlength{\LTcapright}
```

```
5 \newcommand*{\LTcapmarginfalse}{}

```

File 122 **lwarp-ltxtable.sty**

§ 208 Package **ltxtable**

Pkg **ltxtable** **ltxtable** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltxtable}

```
\LTxtable  {\langle width\rangle} {\langle file\rangle}
           2 \newcommand*{\LTxtable}[2]{%
           3 \input{#2}%
           4 }
```

File 123 **lwarp-luacolor.sty**

§ 209 Package **luacolor**

Pkg luacolor luacolor is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{luacolor}

 2 \newcommand{\luacolorProcessBox}[1]{}

File 124 **lwarp-luatodonotes.sty**

§ 210 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 \let\LWRTODONOTES@orig@todototoc\todototoc

```

10
11 \renewcommand*{\todototoc}{%
12 \phantomsection%
13 \LWRTODONOTES@orig@todototoc%
14 }
15
16
17 \renewcommand{\@todonotes@drawMarginNoteWithLine}{%
18 \fcolorbox
19   {\@todonotes@currentbordercolor}
20   {\@todonotes@currentbackgroundcolor}
21   {\arabic{\@todonotes@numberoftodonotes}}
22 \marginpar{\@todonotes@drawMarginNote}
23 }
24
25 \renewcommand{\@todonotes@drawInlineNote}{%
26 \fcolorboxBlock%
27   {\@todonotes@currentbordercolor}%
28   {\@todonotes@currentbackgroundcolor}%
29   {%
30     \if@todonotes@authorgiven%
31     {\@todonotes@author:\,%}
32     \fi%
33     \@todonotes@text%
34   }%
35 }
36
37 \newcommand{\@todonotes@drawMarginNote}{%
38   \if@todonotes@authorgiven%
39     \@todonotes@author\par%
40   \fi%
41   \arabic{\@todonotes@numberoftodonotes}: %
42   \fcolorbox%
43     {\@todonotes@currentbordercolor}%
44     {\@todonotes@currentbackgroundcolor}%
45     {%
46       \@todonotes@sizecommand%
47       \@todonotes@text %
48     }%
49 }%
50
51 \renewcommand{\missingfigure}[2][ ]{%
52 \setkeys{todonotes}{#1}%
53 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
54 \fcolorboxBlock%
55   {\@todonotes@currentbordercolor}%
56   {\@todonotes@currentfigcolor}%
57   {%
58     \setlength{\fboxrule}{4pt}%
59     \fcolorbox{red}{white}{Missing figure} \quad #2%

```

```

60     }
61 }
62
63 \LetLtxMacro\LWRTODONOTES@orig@todocommon\@todocommon
64
65 \RenewDocumentCommand{\@todocommon}{m m}{%
66 \begingroup%
67 \renewcommand*\phantomsection{}%
68 \LWRTODONOTES@orig@todocommon{#1}{#2}%
69 \endgroup%
70 }
71
72 \renewcommand{\@todoarea}[3] [] {%
73     \@todonotes@areaselectedtrue%
74     \@todocommon{#1}{#2}%
75     \@todonotes@textmark@highlight{#3}%
76     \zref@label{\@todonotes@arabic{\@todonotes@numberoftodonotes}@end}%
77 }%
78
79
80 \DeclareDocumentCommand{\@todonotes@textmark@highlight}{m}{%
81 \InlineClass[background:\#B3FFB3]{highlight}{#1}%
82 }
83
84 \fi% \if@todonotes@disabled

```

File 125 **lwarp-marginfit.sty**

§ 211 Package **marginfit**

Pkg marginfit marginfit is ignored.

for HTML output: Discard all options for lwarp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}
```

File 126 **lwarp-marginfix.sty**

§ 212 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*{\marginfix}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1] [] {}
10 \newcommand*{\unblockmargin}[1] [] {}
11 \newcommand*{\marginphantom}[2] [] {}
```

File 127 **lwarp-marginnote.sty**

§ 213 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 128 **lwarp-mcaption.sty**

§ 214 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 129 **lwarp-mdframed.sty**


§ 215 Package **mdframed**

(Emulates or patches code by MARCO DANIEL, ELKE SCHUBERT.)

Pkg mdframed mdframed is loaded with options forced to framemethod=none.

§ 215.1 **Limitations**

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

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

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

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

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

ignored options userdefinedwidth and align are currently ignored.

CSS classes Environments created or encapsulated by mdframed are enclosed in a `<div>` of class `md<environmentname>`, or mdframed otherwise.

Frame titles are placed into a `` 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.

§ 215.2 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}
```

§ 215.3 Patches

Patch to remove PDF formatting and add HTML tags:

```
5 \AtBeginDocument{
6 \def\mdf@trivlist#1{%
7   \edef\mdf@temp{%
8     \topsep=\the\topsep\relax%
9     \partopsep=\the\partopsep\relax%
10    \parsep=\the\parsep\relax%
11  }%
12    \setlength{\topsep}{#1}%
13    \topskip\z@%
14    \partopsep\z@%
15    \parsep\z@%
16    \@nmblistfalse%
17    \@trivlist%
18    \labelwidth\z@%
19    \leftmargin\z@%
20    \itemindent\z@%
21    \let\@itemlabel\@empty%
22    \def\makelabel##1{##1}%
23    \item\relax\mdf@temp\relax%
24  }
25
26 \renewcommand*{\endmdf@trivlist}{%
27 \LWR@traceinfo{endmdf@trivlist}%
28 % \endtrivlist%
29 \LWR@listend%
30 }
31}% \AtBeginDocument
```

§ 215.4 Initial setup

To handle CSS and paragraphs, patch code at start and end of environment and contents. \LWR@origraggedright helps avoid hyphenation.

```
32 \mdfsetup{
33 startcode={\LWR@mdframedstart\LWR@origraggedright},
34 endcode={\LWR@mdframedend},
35 startinnercode={\LWR@startpars\LWR@origraggedright},
36 endinnercode={\LWR@stoppars},
37 }
```

§ 215.5 Color and length HTML conversion

\LWR@mdfprintcolor {\<mdfcolorkey>}

Given the mdframed key, print the color.

```
38 \newcommand*{\LWR@mdfprintcolor}[1]{%
39 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
40 \#\LWR@tempcolor
41 }
```

\LWR@mdfprintlength {\<mdflengthkey>}

Given the mdframed key, print the length.

```
42 \newcommand*{\LWR@mdfprintlength}[1]{%
43 \LWR@printlength{\csuse{mdf@#1@length}}
44 }
```

§ 215.6 Environment encapsulation

\LWR@mdframedstart Actions before an mdframe starts.

Encapsulate a frame inside a <div> of the desired class.

```
45 \newcommand*{\LWR@mdframedstart}{%
46 \LWR@traceinfo{\LWR@mdframedstart start}%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
47 \LWR@stoppars%
```

Open a <div> and with custom class and custom style:

```
48 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
49 style=" \LWR@orignewline
```

Convert and print the background color:

```
50 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```

51 border: \LWR@mdfprintlength{linewidth} solid
52 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline

```

Convert and print the border radius:

```

53 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline

```

Convert and print the shadow:

```

54 \ifbool{mdf@shadow}{%
55     box-shadow:
56     \LWR@mdfprintlength{shadowsize}
57     \LWR@mdfprintlength{shadowsize}
58     \LWR@mdfprintlength{shadowsize}
59     \LWR@mdfprintcolor{shadowcolor} ;
60 }
61 {box-shadow: none ;}
62 \LWR@orignewline

63 "}
64 % \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:

```

65 \LetLtxMacro{\hspace}{\LWR@orighspace}%
66 \LetLtxMacro\rule\LWR@origrule%
67 \LetLtxMacro\makebox\LWR@origmakebox%
68 \LWR@startpars%
69 \LWR@traceinfo{\LWR@mdframedstart done}%
70 }

```

`\LWR@mdframedend` Actions after an mdframe ends.

After closing the <div>, globally restore to the default environment type:

```

71 \newcommand*{\LWR@mdframedend}{
72 \LWR@traceinfo{\LWR@mdframedend start}%

```

Close the custom <div>:

```

73 \LWR@htmldivclassend{\LWR@mdthisenv}

```

Reset future custom class to the default:

```

74 \gdef\LWR@mdthisenv{mdframed}

```

Resume paragraph handling:

```

75 \LWR@startpars%
76 \LWR@traceinfo{\LWR@mdframedend done}%
77 }

```

§ 215.7 **Mdframed environment**

```

78 \renewenvironment{mdframed}[1][ ]{%
79 \color@begingroup%
80 \mdfsetup{userdefinedwidth=\linewidth,#1}%
81 \mdf@startcode%
82 \mdf@preenvsetting%
83 \ifdefempty{\mdf@firstframetitle}{}%
84     {\let\mdf@frametitlesave\mdf@frametitle%
85      \let\mdf@frametitle\mdf@firstframetitle%
86      }%
87 \ifvmode\nointerlineskip\fi%
88     \ifdefempty{\mdf@frametitle}{}%
89     {\mdfframedtitleenv{\mdf@frametitle}%
90      \mdf@@frametitle@use%
91      }%
92 \mdf@trivlist{\mdf@skipabove@length}%%
93 \mdf@settings%
94 % \mdf@lrbox{\mdf@splitbox@one}%
95 % \mdf@startinnercode%
96 }%
97 {%
98 % \mdf@@ignorelastdescenders%
99 \par%
100 % \unskip\ifvmode\nointerlineskip\hrule \@height\z@ \@width\hsize\fi%%
101 \ifmdf@footnoteinside%
102     \def\mdf@reserveda{%
103         \mdf@footnoteoutput%
104         \mdf@endinnercode%
105         \endmdf@lrbox%
106         \ifdefempty{\mdf@frametitle}{}%
107             {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
108             \detected@mdf@put@frame
109         }%
110     \else%
111         \def\mdf@reserveda{%
112             \mdf@endinnercode%
113             \endmdf@lrbox%
114             \ifdefempty{\mdf@frametitle}{}%
115                 {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
116                 \detected@mdf@put@frame%
117             \mdf@footnoteoutput%
118             }%
119         \fi%
120         \mdf@reserveda%
121     \aftergroup\endmdf@trivlist%
122 \color@endgroup%
123 \mdf@endcode%
124 }

```

\mdf@footnoteoutput

```
125 \renewrobustcmd*\mdf@footnoteoutput{%
126   \LWR@printpendingmpfootnotes%
127 }
```

§ 215.8 Titles and subtitles

\mdfframedtitleenv {<title>}

Encapsulation of the original which places the title inside a of class mdfframedtitle:

```
128 \LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv
129
130 \newlength{\LWR@titleroundcorner}
131
132 \renewrobustcmd\mdfframedtitleenv[1]{%
133 \LWR@traceinfo{\LWR@mdfframedtitleenv start}%
134 % \LWR@origmdfframedtitleenv{%
```

Open a with a custom class and custom style:

```
135 \LWR@htmltagc{span class="mdfframedtitle" \LWR@orignewline
136 style=" \LWR@orignewline
```

Convert and print the title background color:

```
137 background:
138 \LWR@mdfprintcolor{frametitlebackgroundcolor}
139 ; \LWR@orignewline
```

Convert and print the title rule:

```
140 \ifbool{mdf@frametitlerule}{%
141   border-bottom:
142   \LWR@mdfprintlength{frametitlerulewidth}
143   solid
144   \LWR@mdfprintcolor{frametitlerulecolor}
145   ; \LWR@orignewline
146 }{}%
```

The title's top border radius is adjusted for the line width:

```
147 border-radius:
148 \setlength{\LWR@titleroundcorner}
149   {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{Opt}}
150 \LWR@printlength{\LWR@titleroundcorner}
151 \LWR@printlength{\LWR@titleroundcorner}
152 Opt Opt
153 \LWR@orignewline
```

Finish the custom style and the opening span tag:

```
154 " \LWR@orignewline
155 }% span
```

Restrict paragraph tags inside a span:

```
156 \begin{LWR@nestspan}%
```

Print the title inside the span:

```
157 #1%
```

Close the span and unnest the paragraph tag restriction:

```
158 \LWR@htmltagc{/span}%
159 \end{LWR@nestspan}%
160 % }
161 \LWR@traceinfo{LWR@mdframedtitleenv end}%
162 }
```

```
\LWR@mdfsubtitlecommon {\langle sub -or- subsub \rangle [\langle options \rangle] {\langle title \rangle}
```

Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubtitle.

Encapsulate the subtitle inside a of class mdframedsubtitle:

```
163 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
164 {% the following empty line is required
165
166 \LWR@traceinfo{LWR@mdframedsubtitlecommon start}%
```

Special handling for mdframed: Subtitles have \pars around them, so temporarily disable them here.

```
167 \let\par\LWR@origpar%
```

Open a with a custom class and custom style:

```
168 \LWR@htmltagc{span class="mdframed#1title"
169 style=" \LWR@orignewline
```

Convert and print the background color:

```
170 background:
171 \LWR@mdfprintcolor{#1titlebackgroundcolor}
172 ; \LWR@orignewline
```

Convert and print the above line:

```
173 \ifbool{mdf@#1titleaboveline}{%
174   border-top:
175   \LWR@mdfprintlength{#1titleabovelinewidth}
176   solid
177   \LWR@mdfprintcolor{#1titleabovelinecolor}
178   ; \LWR@orignewline
179 }{}%
```

Convert and print the below line:

```
180 \ifbool{mdf@#1titlebelowline}{%
181   border-bottom:
```

```

182 \LWR@mdfprintlength{#1titlebelowlinewidth}
183 solid
184 \LWR@mdfprintcolor{#1titlebelowlinecolor}
185 ; \LWR@orignewline
186 }{}%

```

Finish the custom style and the opening span tag:

```

187 "% span

```

Restrict paragraph tags inside a span:

```

188 \begin{LWR@nestspan}%

```

Perform the original subtitle action:

```

189 \IfNoValueTF{#2}
190 {\csuse{LWR@origmdf#1title}{#3}}%
191 {\csuse{LWR@origmdf#1title}[#2]{#3}}%

```

Close the span and unnest the paragraph tag restriction:

```

192 \LWR@htmltagc{/span}% the following empty line is required
193 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
194
195 \LWR@traceinfo{LWR@mdframedsubtitlecommon end}%
196 }

```

```

\LWR@mdfsubtitle  [⟨options⟩] {⟨title⟩}

197 \newcommand*{\LWR@mdfsubtitle}{%
198 \LWR@mdfsubtitlecommon{sub}%
199 }
200 \let\mdfsubtitle\LWR@mdfsubtitle

```

```

\LWR@mdfsubsubtitle  [⟨options⟩] {⟨title⟩}

201 \newcommand*{\LWR@mdfsubsubtitle}{%
202 \LWR@mdfsubtitlecommon{subsub}%
203 }
204 \let\mdfsubsubtitle\LWR@mdfsubsubtitle

```

§ 215.9 New environments

`\LWR@mdthisenv` Stores the environment of the frame about to be created:

```

205 \newcommand*{\LWR@mdthisenv}{mdframed}

```

`\newmdenv` [⟨options⟩] {⟨env-name⟩}

Modified from the original to remember the environment.

```

206 \renewrobustcmd*\newmdenv[2] [] {%
207 \newenvironment{#2}%

```

```

208 {%
209 \mdfsetup{#1}%
210 \renewcommand*{\LWR@mdthisenv}{md#2}%
211 \begin{mdframed}%
212 }
213 {\end{mdframed}}%
214 }

```

`\surroundwithmdframed` [*options*] {*environment*}

Modified from the original to remember the environment.

```

215 \renewrobustcmd*{\surroundwithmdframed}[2] [] {%
216 \BeforeBeginEnvironment{#2}{%
217 \renewcommand*{\LWR@mdthisenv}{md#2}%
218 \begin{mdframed}[#1]}%
219 \AfterEndEnvironment{#2}{\end{mdframed}}}%
220 }

```

`\mdtheorem` [*mdframed-options*] {*envname*} [*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

221 \DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }{%
222 {\ifcsdef{#2}%
223   {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
224   {%
225     \IfNoValueTF {#3}%
226     {%#3 not given -- number relationship
227       \IfNoValueTF {#5}%
228       {%#3+#5 not given
229         \@definecounter{#2}%
230         \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
231         \newenvironment{#2}[1] [] {%
232           \refstepcounter{#2}%
233           \ifstrempy{##1}%
234             {\let\@temptitle\relax}%
235             {%
236               \def\@temptitle{\mdf@theoremseparator%
237                 \mdf@theoremspace%
238                 \mdf@theoremtitlefont%
239                 ##1}%
240               \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
241             }%
242             \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
243               \@temptitle}]]%
244             {\end{mdframed}}%
245           \newenvironment{#2*}[1] [] {%
246             \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
247             \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%

```

```

248         {\end{mdframed}}}%
249     }%
250     {%#5 given -- reset counter
251     \@definecounter{#2}\@newctr{#2}[#5]%
252     \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
253     \expandafter\xdef\csname the#2\endcsname{%
254         \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
255         \@thmcounter{#2}}%
256     \newenvironment{#2}[1][]{%
257         \refstepcounter{#2}%
258         \ifstrepty{##1}%
259             {\let\@temptitle\relax}%
260             {%
261                 \def\@temptitle{\mdf@theoremseparator%
262                     \mdf@theoremspace%
263                     \mdf@theoremtitlefont%
264                     ##1}%
265                 \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
266             }
267     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
268         \@temptitle}]]%
269         {\end{mdframed}}}%
270     \newenvironment{#2*}[1][]{%
271         \ifstrepty{##1}%
272             {\let\@temptitle\relax}%
273             {%
274                 \def\@temptitle{\mdf@theoremseparator%
275                     \mdf@theoremspace%
276                     \mdf@theoremtitlefont%
277                     ##1}%
278                 \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
279             }%
280     \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
281         {\end{mdframed}}}%
282     }%
283 }%
284 {%#3 given -- number relationship
285 \global\@namedef{the#2}{\@nameuse{the#3}}%
286 \newenvironment{#2}[1][]{%
287     \refstepcounter{#3}%
288     \ifstrepty{##1}%
289         {\let\@temptitle\relax}%
290         {%
291             \def\@temptitle{\mdf@theoremseparator%
292                 \mdf@theoremspace%
293                 \mdf@theoremtitlefont%
294                 ##1}%
295             \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
296         }
297     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%

```

```

298                                     \@temptitle}}}%
299         {\end{mdframed}}}%
300     \newenvironment{#2*}[1][]{%
301         \ifstrepty{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
302         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}}}%
303         {\end{mdframed}}}%
304     }%
305     \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}% lwarp
306     \BeforeBeginEnvironment{#2*}{\renewcommand*\LWR@mdthisenv}{md#2}}% lwarp
307 }%
308 }

```

`\newmdtheoremenv` [*mdframed-options*] {*envname*} [*numberedlike*] {*caption*} [*within*]

Modified from the original to remember the environment.

```

309 \DeclareDocumentCommand\newmdtheoremenv{0}{m o m o }{%
310   \ifbool{test}{\IfNoValueTF{#3}} and test {\IfNoValueTF{#5}} }%
311   {\newtheorem{#2}{#4}}%
312   {%
313     \IfValueT{#3}{\newtheorem{#2}{#3}{#4}}%
314     \IfValueT{#5}{\newtheorem{#2}{#4}{#5}}%
315   }%
316   \BeforeBeginEnvironment{#2}{%
317     \renewcommand*\LWR@mdthisenv}{md#2}%
318     \begin{mdframed}[#1]}%
319   \AfterEndEnvironment{#2}{%
320     \end{mdframed}}%
321 }

```

File 130 **lwarp-memhfixc.sty**

§ 216 Package **memhfixc**

Pkg `memhfixc` `memhfixc` is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{memhfixc}

File 131 **lwarp-metalogo.sty**

§ 217 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 132 **lwarp-microtype.sty**

§ 218 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{\lslig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1#{\@gobble}
26 \@onlypreamble\DeclareMicrotypeSet

```

```

27 \@onlypreamble\UseMicrotypeSet
28 \@onlypreamble\DeclareMicrotypeSetDefault
29 \@onlypreamble\DisableLigatures
30 \@onlypreamble\DeclareMicrotypeVariants
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

File 133 **lwarp-midfloat.sty**

§ 219 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 134 **lwarp-midpage.sty**

§ 220 Package **midpage**

Pkg midpage midpage is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{midpage}

2 \newenvironment{midpage}

3 {\begin{BlockClass}[\LWR@origmbox{margin-top:6ex} ; \LWR@origmbox{margin-bottom:6ex}]{midpage}}

4 {\end{BlockClass}}

File 135 **lwarp-moreverb.sty**

§ 221 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{verbatim}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{verbatim}{%
8 \LWR@afterendverbatim%
9 }
10
11
12 \LetLtxMacro\LWRMV@orig@verbatiminput\@verbatiminput
13
14 \renewcommand{\@verbatiminput}[2][{}]{%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatiminput[#1]{#2}%
18 \LWR@afterendverbatim%
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
23 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 % \unskip\LWR@origvspace*{-\baselineskip}%
28 \LWR@afterendverbatim%
29 }
30
31 \BeforeBeginEnvironment{listingcont}{%
32 \LWR@forcenewpage
33 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
34 }
35
36 \AfterEndEnvironment{listingcont}{%
37 % \unskip\LWR@origvspace*{-\baselineskip}%
38 \LWR@afterendverbatim%
39 }

40 \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@afterendverbatim%
47 }
48

```

63 \end{warpHTML}

lwap-morerevrites.sty

morewrites

Pkg	morewrites	Error if morewrites is loaded after lwarp.
-----	------------	--

Discard all options for lwrap-morewrites:

for HTML output: `1 \LWR@ProvidesPackageDrop{morewrites}`

```
2 \LWR@loadbefore{morewrites}
```

lwrap-mparhack.sty

mparhack

Pkg mparhack Ignored.

for HTML output: Discard all options for lwrap-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

File 138 **lwarp-multicol.sty**

§ 224 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 139 **lwarp-multirow.sty**

§ 225 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 63.19 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}{}

```

§ 225.1 **Multirow**

`\multirow` [*vpos*] {*numrows*} [*bigstruts*] {*width*} [*fixup*] {*text*}

```

4 \RenewDocumentCommand{\multirow}{O{c} m o 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="\LWR@origmbox{vertical-align:bottom}" }{}%
12 \ifstrequal{#1}{t}{style="\LWR@origmbox{vertical-align:top}" }{}%
13 }%
```

The left/right border, if given:

```
14 \ifdefvoid{\LWR@multirowborder}{}{%
15 style="\LWR@origmbox{border-\LWR@multirowborder:} 2px dotted black ; %
16 \LWR@origmbox{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@addtabularrulecolors%
27 \LWR@tdendstyles%
28 }%
```

The column's < spec:

```
29 \LWR@getexparray{\LWR@colbeforespec}{\arabic{\LWR@tablecolindex}}%
```

While printing the text, redefine \ to generate a new line

```
30 \begingroup\LetLtxMacro{\}{\LWR@endofline}\#6\endgroup%
31 \LWR@stoppars%
32 \global\boolfalse{\LWR@intabularmetadata}%
```

```

33 \renewcommand{\LWR@multirowborder}{}%
34 \LWR@traceinfo{*** multirow done}%
35 }%

```

§ 225.2 Combined multicolumn and multirow

⚠ `\multicolumn` &
`\multirow`

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

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

The two arguments for `\multicolumn` come first, followed by the five arguments for `\multirow`, many of which are optional, followed by the contents.

⚠ **skipped cells**

As per `\multirow`, skipped cells to the right of the `\multicolumnrow` statement are not included in the source code on the same line. On the following lines,

⚠ **empty cells**

`\mcolrowcell` must be used for each cell of each column and each row to be skipped:

```

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

```

vposn

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

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

```
36 \NewDocumentCommand{\multicolumnrow}{m m O{} m O{} m O{} +m}{%
```

Figure out how many extra HTML columns to add for @ and ! columns:

```
37 \LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}
```

Create the multicolumn/multirow tag:

```

38 \begingroup%
39 \LetLtxMacro{\}\{\LWR@endofline}%
40 \LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
41 \endgroup%

```

Move to the next \LaTeX column:

```

42 \addtocounter{LWR@tablecolindex}{#1}%
43 \addtocounter{LWR@tablecolindex}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```
44 \booltrue{LWR@skipatbang}%
45 }

46 \appto{LWR@restoreorigformatting}{%
47 \LetLtxMacro\multirow\LWR@origmultirow%
48 \renewcommand{\multicolumnrow}{LWR@origmulticolumnrow}%
49 }
```

File 140 **lwarp-multitoc.sty**

§ 226 Package **multitoc**

Pkg multitoc multitoc is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{multitoc}

2 \newcommand{\multicolumntoc}{2}
3 \newcommand{\multicolumnlot}{2}
4 \newcommand{\multicolumnlof}{2}
5 \newcommand*{\immediateaddtocontents}[2]{}
```

File 141 **lwarp-nameref.sty**

§ 227 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 142 **lwarp-natbib.sty**

§ 228 Package **natbib**

(Emulates or patches code by PATRICK W. DALY.)

Pkg natbib natbib is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{natbib}

Replace math $<$ and $>$ with `\textless` and `\textgreater`:

A macro to compare:

```
2 \newcommand{\LWRNB@NAT@open}{${<}
```

To patch `\NAT@open` and `\NAT@close`

```
3 \newcommand{\LWRNB@patchnatbibopenclose}{
4 \ifdefstrequal{\NAT@open}{\LWRNB@NAT@open}
5 {
6   \renewcommand{\NAT@open}{\textless}
7   \renewcommand{\NAT@close}{\textgreater}
8 }{}
9 }
```

Do it now in case angle was selected as an option:

```
10 \LWRNB@patchnatbibopenclose
```

Also patch `\setcitestyle` to patch after settings are made:

```
11 \let\LWRNB@origsetcitestyle\setcitestyle
12
13 \renewcommand{\setcitestyle}[1]{%
14 \LWRNB@origsetcitestyle{#1}%
15 \LWRNB@patchnatbibopenclose%
16 }
```

File 143 **lwarp-needspace.sty**

§ 229 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 \DeclareDocumentCommand{\needspace}{m}{}
4 \DeclareDocumentCommand{\Needspace}{s m}{}

```

File 144 **lwarp-newclude.sty**

§ 230 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 145 **lwarp-newunicodechar.sty**

§ 231 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 146 **lwarp-nextpage.sty**

§ 232 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 \DeclareDocumentCommand{\cleartoevenpage}{o}{}
 3 \DeclareDocumentCommand{\movetoevenpage}{o}{}
 4 \DeclareDocumentCommand{\cleartooddpage}{o}{}
 5 \DeclareDocumentCommand{\movetooddpage}{o}{}
 6 \end{document}

File 147 **lwarp-nonumonpart.sty**

§ 233 Package **nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

File 148 **lwarp-nopageno.sty**

§ 234 Package **nopageno**

Pkg nopageno nopageno is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

File 149 **lwarp-nowidow.sty**

§ 235 Package **nowidow**

(Emulates or patches code by RAPHAËL PINSON.)

Pkg nowidow nowidow is not used during HTML conversion.

Discard all options for lwarp-nowidow:

for HTML output: 1 \LWR@ProvidesPackageDrop{nowidow}

 \nowidow [*<lines>*]
 \setnowidow [*<lines>*]
 2 \newcommand*{\nowidow}[1] [] {}
 3 \newcommand*{\setnowidow}[1] [] {}

 \noclub [*<lines>*]
 \setnoclub [*<lines>*]
 4 \newcommand*{\noclub}[1] [] {}
 5 \newcommand*{\setnoclub}[1] [] {}

File 150 **lwarp-nttheorem.sty**

§ 236 Package **nttheorem**

(Emulates or patches code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg nttheorem nttheorem 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.

§ 236.1 **Limitations**

⚠ **Font control** This conversion is not total. Font control is via CSS, and the custom \TeX font settings are ignored.

⚠ **Equation numbering** nttheorem 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 nttheorem's `thref` option.

§ 236.2 **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}

```

§ 236.3 Remembering the theorem style

Storage for the style being used for new theorems.

```

24 \newcommand{\LWR@newtheoremstyle}{plain}

25 \AtBeginDocument{
26 \@ifpackageloaded{cleveref}{
27 \gdef\@thm#1#2#3{%
28   \if@thmmarks
29     \stepcounter{end\InTheoType ctr}%
30   \fi
31   \renewcommand{\InTheoType}{#1}%
32   \if@thmmarks
33     \stepcounter{curr#1ctr}%
34     \setcounter{end#1ctr}{0}%
35   \fi
36   \refstepcounter[#1]{#2}% <<< cleveref modification
37   \theorem@prework
38   \LWR@forcenewpage% lwarp
39   \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
40   \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
41   \ifuse@newframeskips % cf. latex.ltx for topsepadd: \@trivlist
42     \ifthm@inframe
43       \thm@topsep\theoreminframepreskipamount
44       \thm@topsepadd\theoreminframepostskipamount
45     \else
46       \thm@topsep\theorempreskipamount

```

```

47     \thm@topsepadd\theorempostskipamount
48     \fi
49     \else% oldframeskips
50     \thm@topsep\theorempreskipamount
51     \thm@topsepadd \theorempostskipamount
52     \ifvmode\advance\thm@topsepadd\partopsep\fi
53     \fi
54     \@topsep\thm@topsep
55     \@topsepadd\thm@topsepadd
56     \advance\linewidth -\theorem@indent
57     \advance\linewidth -\theorem@rightindent
58     \advance\@totalleftmargin \theorem@indent
59     \parshape \@ne \@totalleftmargin \linewidth
60     \ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
61 }
62 }{% not @ifpackageloaded{cleveref}
63 \gdef\@thm#1#2#3{%
64     \if@thmmarks
65         \stepcounter{end\InTheoType ctr}%
66     \fi
67     \renewcommand{\InTheoType}{#1}%
68     \if@thmmarks
69         \stepcounter{curr#1ctr}%
70         \setcounter{end#1ctr}{0}%
71     \fi
72     \refstepcounter{#2}%
73     \theorem@prework
74     \LWR@forcenewpage% lwarp
75     \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
76     \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
77     \ifuse@newframeskips % cf. latex.ltx for topsepadd: \@trivlist
78     \ifthm@inframe
79         \thm@topsep\theoreminframepreskipamount
80         \thm@topsepadd\theoreminframepostskipamount
81     \else
82         \thm@topsep\theorempreskipamount
83         \thm@topsepadd\theorempostskipamount
84     \fi
85     \else% oldframeskips
86     \thm@topsep\theorempreskipamount
87     \thm@topsepadd \theorempostskipamount
88     \ifvmode\advance\thm@topsepadd\partopsep\fi
89     \fi
90     \@topsep\thm@topsep
91     \@topsepadd\thm@topsepadd
92     \advance\linewidth -\theorem@indent
93     \advance\linewidth -\theorem@rightindent
94     \advance\@totalleftmargin \theorem@indent
95     \parshape \@ne \@totalleftmargin \linewidth
96     \ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}

```

```

97 }
98 }
99 }% AtBeginDocument

```

Patched to remember the style being used for new theorems:

```

100 \gdef\theoremstyle#1{%
101   \ifundefined{th@#1}{\@warning
102     {Unknown theoremstyle '#1'. Using 'plain'}}%
103     \theoremstyle{plain}
104     \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
105   }%
106   {
107     \theoremstyle{#1}
108     \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
109   }
110 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

111
112 \gdef\xnthm#1#2[#3]{%
113   \ifthm@tempif
114     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
115     \expandafter\ifundefined{c@#1}%
116       {\@definecounter{#1}}{}%
117     \@newctr{#1}[#3]%
118     \expandafter\xdef\csname the#1\endcsname{%
119       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
120       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
121     \expandafter\gdef\csname mkheader@#1\endcsname
122       {\csname setparams@#1\endcsname
123        \@thm{#1}{#1}{#2}
124       }%
125     \global\@namedef{end#1}{\@endtheorem}
126     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#1}}}% lwarp
127   \fi
128 }
129
130 \gdef\ynthm#1#2{%
131   \ifthm@tempif
132     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
133     \expandafter\ifundefined{c@#1}%
134       {\@definecounter{#1}}{}%
135     \expandafter\xdef\csname the#1\endcsname
136       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
137     \expandafter\gdef\csname mkheader@#1\endcsname
138       {\csname setparams@#1\endcsname

```

```

139      \@thm{#1}{#1}{#2}
140      }%
141      \global\@namedef{end#1}{\@endtheorem}
142      \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
143      \fi
144  }
145
146  \gdef\@othm#1[#2]#3{%
147    \ifundefined{c@#2}{\@nocounterr{#2}}%
148    {\ifthm@tempif
149      \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
150      \global\@namedef{the#1}{\@nameuse{the#2}}%
151      \expandafter\protected@xdef\csname num@addtheorem#1\endcsname{%
152        \noexpand\@num@addtheorem#1}{#3}}%
153      \expandafter\protected@xdef\csname nonum@addtheorem#1\endcsname{%
154        \noexpand\@nonum@addtheorem#1}{#3}}%
155      \theoremkeyword{#3}%
156      \expandafter\protected@xdef\csname #1Keyword\endcsname
157        {\the\theoremkeyword}%
158      \expandafter\gdef\csname mkheader@#1\endcsname
159        {\csname setparms@#1\endcsname
160          \@thm{#1}{#2}{#3}
161          }%
162      \global\@namedef{end#1}{\@endtheorem}
163      \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
164      \fi}
165  }

```

§ 236.4 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.

```

166 \newcommand{\LWR@inctheorem}{%
167 \addtocounter{LWR@thisautoid}{1}%
168 \LWR@stoppars%
169 \LWR@htmltag{a id="\LWR@origmbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
170 \LWR@startpars%
171 }

```

§ 236.5 \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>`.

```

172 \gdef\newtheoremstyle#1#2#3{%
173   \expandafter\@ifundefined{th@#1}%
174   {\expandafter\gdef\csname th@#1\endcsname{%
175     \def\@begintheorem####1####2{%
176       \LWR@intheorem% lwarp
177       #2}%
178     \def\@opargbegintheorem####1####2####3{%
179       \LWR@intheorem% lwarp
180       #3}%
181   }%
182 }%
183 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
184 }
```

§ 236.6 Standard styles

```

185 \renewtheoremstyle{plain}%
186   {\item[
187     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
188   {\item[
189     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}
190
191 \renewtheoremstyle{break}%
192   {\item[
193     \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
194     ]}%
195   {\item[
196     \InlineClass{theoremheaderbreak}%
197     {##1\ ##2\ (##3)\theorem@separator}\newline
198     ]}
199
200 \renewtheoremstyle{change}%
201   {\item[
202     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]]}%
203   {\item[
204     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]]}
205
206 \renewtheoremstyle{changebreak}%
207   {\item[
208     \InlineClass{theoremheaderchangebreak}%
209     {##2\ ##1\theorem@separator}\newline
210     ]}%
211   {\item[
212     \InlineClass{theoremheaderchangebreak}%
213     {##2\ ##1\ (##3)\theorem@separator}\newline
214     ]}
215
```

```

216 \renewtheoremstyle{margin}%
217   {\item[
218     \InlineClass{theoremheadermargin}{##2 \quad ##1\theoremseparator}
219   ]}%
220   {\item[
221     \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theoremseparator}
222   ]}
223
224 \renewtheoremstyle{marginbreak}%
225   {\item[
226     \InlineClass{theoremheadermarginbreak}%
227       {##2 \quad ##1\theoremseparator}\newline
228   ]}%
229   {\item[
230     \InlineClass{theoremheadermarginbreak}%
231       {##2 \quad ##1\ (##3)\theoremseparator}\newline
232   ]}
233
234 \renewtheoremstyle{nonumberplain}%
235   {\item[
236     \InlineClass{theoremheaderplain}{##1\theoremseparator}]]}%
237   {\item[
238     \InlineClass{theoremheaderplain}{##1\ (##3)\theoremseparator}]]}
239
240 \renewtheoremstyle{nonumberbreak}%
241   {\item[
242     \InlineClass{theoremheaderbreak}{##1\theoremseparator}\newline
243   ]}%
244   {\item[
245     \InlineClass{theoremheaderbreak}{##1\ (##3)\theoremseparator}\newline
246   ]}
247
248 \renewtheoremstyle{empty}%
249   {\item[]}%
250   {\item[
251     \InlineClass{theoremheaderplain}{##3}]]}
252
253 \renewtheoremstyle{emptybreak}%
254   {\item[]}%
255   {\item[
256     \InlineClass{theoremheaderplain}{##3} \ \newline}

```

§ 236.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

257 \ifbool{LWR@theoremamsthm}{-}{%

```

Upright text via CSS:

```

258 \newtheoremstyle{plainupright}%
259 {\item[
260 \InlineClass{theoremheaderplain}{##1\ ##2\theoremseparator}]]}%
261 {\item[
262 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theoremseparator}]]}

```

Upright text and small caps header via CSS:

```

263 \newtheoremstyle{nonumberplainuprightsc}%
264 {\item[
265 \InlineClass{theoremheadersc}{##1\theoremseparator}]]}%
266 {\item[
267 \InlineClass{theoremheadersc}{##1\ (##3)\theoremseparator}]]}
268 }% not amsthm

```

§ 236.8 Renewed standard configuration

The following standard configuration is renewed using the new css:

```

269 \ifbool{LWR@ntheoremamsthm}{-}{%
270 \theoremnumbering{arabic}
271 \theoremstyle{plain}
272 \RequirePackage{latexsym}
273 \theoremsymbol{\Box}
274 \theorembodyfont{\itshape}
275 \theoremheaderfont{\normalfont\bfseries}
276 \theoremseparator{}
277 \renewtheorem{Theorem}{Theorem}
278 \renewtheorem{theorem}{Theorem}
279 \renewtheorem{Satz}{Satz}
280 \renewtheorem{satz}{Satz}
281 \renewtheorem{Proposition}{Proposition}
282 \renewtheorem{proposition}{Proposition}
283 \renewtheorem{Lemma}{Lemma}
284 \renewtheorem{lemma}{Lemma}
285 \renewtheorem{Korollar}{Korollar}
286 \renewtheorem{korollar}{Korollar}
287 \renewtheorem{Corollary}{Corollary}
288 \renewtheorem{corollary}{Corollary}
289
290 \theoremstyle{plainupright}
291 \theorembodyfont{\upshape}
292 \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
293 \renewtheorem{Example}{Example}
294 \renewtheorem{example}{Example}
295 \renewtheorem{Beispiel}{Beispiel}
296 \renewtheorem{beispiel}{Beispiel}

```

```

297 \renewtheorem{Bemerkung}{Bemerkung}
298 \renewtheorem{bemerkung}{Bemerkung}
299 \renewtheorem{Anmerkung}{Anmerkung}
300 \renewtheorem{anmerkung}{Anmerkung}
301 \renewtheorem{Remark}{Remark}
302 \renewtheorem{remark}{Remark}
303 \renewtheorem{Definition}{Definition}
304 \renewtheorem{definition}{Definition}
305
306 \theoremstyle{nonumberplainuprightsc}
307 \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
308 \renewtheorem{Proof}{Proof}
309 \renewtheorem{proof}{Proof}
310 \renewtheorem{Beweis}{Beweis}
311 \renewtheorem{beweis}{Beweis}
312 \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
313
314 \theoremsymbol{}
315}% not amsthm

```

§ 236.9 amsthm option

Only if the amsthm option was given:

```

316 \ifbool{LWR@theoremamsthm}{
317
318 \gdef\th@plain{%
319 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
320 \def\@begintheorem##1##2{%
321 \LWR@intheorem% lwarp
322 \item[
323 \InlineClass{theoremheaderplain}{##1\ ##2.}
324 ]}%
325 \def\@opargbegintheorem##1##2##3{%
326 \LWR@intheorem% lwarp
327 \item[
328 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
329 ]}}
330
331 \gdef\th@nonumberplain{%
332 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
333 \def\@begintheorem##1##2{%
334 \LWR@intheorem% lwarp
335 \item[
336 \InlineClass{theoremheaderplain}{##1.}
337 ]}%
338 \def\@opargbegintheorem##1##2##3{%
339 \LWR@intheorem% lwarp
340 \item[

```

```

341 \InlineClass{theoremheaderplain}{##1\ (##3).}
342   }}
343
344 \gdef\th@definition{%
345   \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
346   \def\@begintheorem##1##2{%
347     \LWR@intheorem% lwarp
348     \item[
349   \InlineClass{theoremheaderdefinition}{##1\ ##2.}
350   ]}%
351   \def\@opargbegintheorem##1##2##3{%
352     \LWR@intheorem% lwarp
353     \item[
354   \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
355   ]}}
356
357 \gdef\th@nonumberdefinition{%
358   \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
359   \def\@begintheorem##1##2{%
360     \LWR@intheorem% lwarp
361     \item[
362   \InlineClass{theoremheaderdefinition}{##1.}
363   ]}%
364   \def\@opargbegintheorem##1##2##3{%
365     \LWR@intheorem% lwarp
366     \item[
367   \InlineClass{theoremheaderdefinition}{##1\ (##3).}
368   ]}}
369
370 \gdef\th@remark{%
371   \def\theorem@headerfont{\itshape}\normalfont%
372   \def\@begintheorem##1##2{%
373     \LWR@intheorem% lwarp
374     \item[
375   \InlineClass{theoremheaderremark}{##1\ ##2.}
376   ]}%
377   \def\@opargbegintheorem##1##2##3{%
378     \LWR@intheorem% lwarp
379     \item[
380   \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
381   ]}}
382
383 \gdef\th@nonumberremark{%
384   \def\theorem@headerfont{\itshape}\normalfont%
385   \def\@begintheorem##1##2{%
386     \LWR@intheorem% lwarp
387     \item[
388   \InlineClass{theoremheaderremark}{##1.}
389   ]}%
390   \def\@opargbegintheorem##1##2##3{%

```

```

391      \LWR@inctheorem% lwarp
392      \item[
393 \InlineClass{theoremheaderremark}{##1\ (##3).}
394 ]}]
395
396 \gdef\th@proof{%
397 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
398 \def\@begintheorem##1##2{%
399 \LWR@inctheorem% lwarp
400 \item[
401 \InlineClass{theoremheaderproof}{##1.}
402 ]}%
403 \def\@opargbegintheorem##1##2##3{%
404 \LWR@inctheorem% lwarp
405 \item[
406 \InlineClass{theoremheaderproof}{##1\ (##3).}
407 ]}]
408
409
410
411 \newcounter{proof}%
412 \if@thmmarks
413 \newcounter{currproofctr}%
414 \newcounter{endproofctr}%
415 \fi
416
417 \gdef\proofSymbol{\openbox}
418
419 \newcommand{\proofname}{Proof}
420
421 \newenvironment{proof}[1][\proofname]{
422 \th@proof
423 \def\theorem@headerfont{\itshape}%
424 \normalfont
425 \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
426 \@thm{proof}{proof}{#1}
427 }%
428 {\@endtheorem}
429
430 }{}% amsthm option

```

§ 236.10 Ending a theorem

Patched for CSS:

```

431 \let\LWR@origendtheorem\@endtheorem
432 \renewcommand{\@endtheorem}{%
433 \ifbool{LWR@nththeoremmarks}{%
434   \ifsetendmark%
435   \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
436   \setendmarkfalse%
437   \fi%
438 }{}%
439 \LWR@origendtheorem% also does \@endtrivlist
440 \ifbool{LWR@nththeoremmarks}{\global\setendmarktrue}{}%
441 \endBlockClass%
442 }
```

§ 236.11 \NoEndMark

```

443 \gdef\NoEndMark{\global\setendmarkfalse}
```

§ 236.12 List-of

Redefined to reuse the float mechanism to add list-of-theorem links:

```
\thm@thmline {\langle 1: printed type \rangle} {\langle 2: # \rangle} {\langle 3: optional \rangle} {\langle 4: page \rangle}
```

```

444 \renewcommand{\thm@@thmline@noname}[4]{%
445 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
446 }
447
448 \renewcommand{\thm@@thmline@name}[4]{%
449 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
450 }
```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```

451 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for CSS:

```

452 \def\listtheorems#1{
453 \LWR@htmlclass{nav}{lothm}%
454 \begingroup
455 \c@tocdepth=-2%
456 \def\thm@list{#1}\thm@processlist
457 \endgroup
458 \LWR@htmlclassend{nav}{lothm}%
459 }
```

§ 236.13 Symbols

Proof QED symbol:

```
460 \newcommand{\qed}{\quad\the\qedsymbol}
461
462 \AtBeginDocument{
463 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
464 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
465 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
466 }
```

§ 236.14 Cross-referencing

`\thref {<label>}`

```
467 \newcommand*{\thref}[1]{\cref{#1}}
```

File 151 lwarp-overpic.sty

§ 237 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 75.2 for the print-mode version of `\overpicfontsize` and `\overpicfontskip`.

for HTML output:

```
1 \LWR@ProvidesPackagePass{overpic}

2 \newcommand*{\overpicfontsize}{12}
3 \newcommand*{\overpicfontskip}{14}
4
5 \BeforeBeginEnvironment{overpic}{%
6   \begin{lateximage}%
7   \fontsize{\overpicfontsize}{\overpicfontskip}%
8   \selectfont%
9 }
10
11 \AfterEndEnvironment{overpic}{\end{lateximage}}
12
13 \BeforeBeginEnvironment{Overpic}{%
```

```

14 \begin{lateximage}%
15 \fontsize{\overpicfontsize}{\overpicfontskip}%
16 \selectfont%
17 }
18
19 \AfterEndEnvironment{Overpic}{\end{lateximage}}

```

File 152 **lwarp-pagegrid.sty**

§ 238 Package **pagegrid**

Pkg pagegrid pagegrid is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{pagegrid}

2 \newcommand*{\pagegridsetup}[1]{}
```

File 153 **lwarp-pagenote.sty**

§ 239 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 154 **lwarp-pagesel.sty**

§ 240 Package **pagesel**

Pkg pagesel pagesel is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{pagesel}
```

File 155 **lwarp-paralist.sty**

§ 241 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 156 **lwarp-parskip.sty**

§ 242 Package **parskip**

Pkg parskip parskip is ignored.

for HTML output: Discard all options for lwarp-parskip.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

File 157 **lwarp-pdfscape.sty**

§ 243 Package **pdfscape**

Pkg pdfscape pdfscape is ignored.

for HTML output: Discard all options for lwarp-pdfscape:

```
1 \LWR@ProvidesPackageDrop{pdfscape}
```

File 158 **lwarp-pdfrender.sty**

§ 244 Package **pdfrender**

Pkg pdfrender pdfrender is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{pdfrender}

2 \newcommand*{\pdfrender}[1]{}
3 \newcommand{\textpdfrender}[2]{#2}

```

File 159 **lwarp-pdfsync.sty**

§ 245 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 160 **lwarp-pfnote.sty**

§ 246 Package **pfnote**

Pkg pfnote pfnote is emulated.

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

for HTML output:

```

1 \LWR@ProvidesPackageDrop{pfnote}

```

File 161 **lwarp-placeins.sty**

§ 247 Package **placeins**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg placeins placeins is not used during HTML conversion.

Discard all options for lwarp-placeins:

for HTML output: 1 \LWR@ProvidesPackageDrop{placeins}

 2 \newcommand*{\FloatBarrier}{}

File 162 **lwarp-prelim2e.sty**

§ 248 Package **prelim2e**

(Emulates or patches code by MARTIN SCHRÖDER.)

Pkg prelim2e Emulated.

for HTML output: Discard all options for lwarp-prelim2e:

 1 \LWR@ProvidesPackageDrop{prelim2e}

 2 \newcommand{\PrelimText}{}

 3 \newcommand{\PrelimTextStyle}{}

 4 \newcommand{\PrelimWords}{}

File 163 **lwarp-prettyref.sty**

§ 249 Package **prettyref**

(Emulates or patches code by KEVIN S. RULAND.)

Pkg prettyref prettyref is patched for use by lwarp.

for HTML output: 1 \LWR@ProvidesPackagePass{prettyref}

```
2 \newreformat{fig}{Figure \ref{#1}}
3 \newreformat{tab}{Table \ref{#1}}
```

File 164 **lwarp-preview.sty**

§ 250 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 165 **lwarp-quotchap.sty**

§ 251 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 \begin{BlockClass}[max-width: \LWR@printlength{\LWR@quotchapwidth}]{quotchap}
9 %\begin{minipage}{\LWR@quotchapwidth}
10 \@quotchap
11 %\end{minipage}
12 \end{BlockClass}
```

Deactivate the quote printing:

```
13 \global\let\@printcites\relax
14 }
15
16 \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>

```
17 \setlength{\LWR@quotchapwidth}{#1*2}%
18 \global\LWR@quotchapwidth=\LWR@quotchapwidth%
```

Remember the body, and activate the quote printing:

```
19 \global\let\@quotchap\BODY
20 \global\let\@printcites\@iprintcites%
21 }
```

The quotation author is placed inside a <div> of class qauthor:

```
22 \newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

\qsetcnfont is ignored:

```
23 \newcommand{\qsetcnfont}[1]{}
```

File 166 **lwarp-ragged2e.sty**

§ 252 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 167 **lwarp-realscripts.sty**

§ 253 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}{\LWR@origmbox{text-align:center}}
15 \ifthenelse{\equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{\LWR@origmbox{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 168 **lwarp-reldsize.sty**

§ 254 Package **reldsize**


(Emulates or patches code by DONALD ARSENEAU, BERNIE COSELL, MATT SWIFT.)

Pkg **reldsize** reldsize 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.

`\reldsize`, `\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{reldsize}

2 \let\LWR@origreldsize\reldsize
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\reldsize\LWR@origreldsize%
12 \LetLtxMacro\larger\LWR@origlarger%
13 \LetLtxMacro\smaller\LWR@origsmaller%
14 \let\relscale\LWR@relscale%
15 \LetLtxMacro\textlarger\LWR@origtextlarger%
16 \LetLtxMacro\textsmaller\LWR@origtextsmaller%

```

```

17 \let\textscale\LWR@textscale%
18 }
19
20 \newcounter{LWR@relsizetemp}
21
22 \renewcommand*{\relsize}[1]{%
23 \renewcommand*{\larger}[1][1]{%
24 \renewcommand*{\smaller}[1][1]{%
25 \renewcommand*{\relscale}[1]{%
26
27 \renewcommand*{\textlarger}[2][1]{%
28 \setcounter{LWR@relsizetemp}{100+(#1*20)}%
29 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textlarger}{#2}%
30 }
31
32 \renewcommand*{\textsmaller}[2][1]{%
33 \setcounter{LWR@relsizetemp}{100-(#1*20)}%
34 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textsmaller}{#2}%
35 }
36
37 \renewcommand*{\textscale}[2]{%
38 \setcounter{LWR@relsizetemp}{100*\real{#1}}%
39 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textscale}{#2}%
40 }

```

File 169 **lwarp-resizegather.sty**

§ 255 Package **resizegather**

Pkg `resizegather` `resizegather` is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{resizegather}

 2 \newcommand*{\resizegathersetup}[1]{}

File 170 **lwarp-romanbar.sty**

§ 256 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 171 **lwarp-romanbarpagenumber.sty**

§ 257 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{romanbarpagenumber}
```

File 172 **lwarp-rotating.sty**

§ 258 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 173 **lwarp-rotfloat.sty**

§ 259 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 {<1: type>} {<2: placement>} {<3: ext>} [<4: within>]

Emulates the \newfloat command from the float package. Sideways floats are \let to the same as regular floats.

“placement” is ignored.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}%
6 {%
7   \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
8 }%
9 {%
10  \DeclareFloatingEnvironment[fileext=#3]{#1}%

11  \DeclareFloatingEnvironment[fileext=#3]{sideways#1}%
12 }%
13 \csletcs{sideways#1}{#1}%
14 \csletcs{endsideways#1}{end#1}%

```

newfloat package automatically creates the \listof command for new floats, but float does not, so remove \listof here in case it is manually created later.

```

15 \cslet{listof#1s}\relax%
16 \cslet{listof#1es}\relax%
17 }

```

File 174 **lwarp-savetrees.sty**

§ 260 Package **savetrees**

Pkg **savetrees** Emulated.

for HTML output: Discard all options for lwarp-savetrees:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

File 175 **lwarp-scalefnt.sty**

§ 261 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 176 **lwarp-scrextend.sty**

§ 262 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{\cleardoubleoddpager}{}{}
35 \DeclareDocumentCommand{\cleardoubleoddpagerusingstyle}{m}{}
36 \DeclareDocumentCommand{\cleardoubleoddpageremptypage}{}{}
37 \DeclareDocumentCommand{\cleardoubleoddpagerplainpage}{}{}
38 \DeclareDocumentCommand{\cleardoubleoddpagerstandardpage}{}{}
39 \DeclareDocumentCommand{\cleardoubleevenpage}{}{}
40 \DeclareDocumentCommand{\cleardoubleevenpagerusingstyle}{m}{}
41 \DeclareDocumentCommand{\cleardoubleevenpageremptypage}{}{}
42 \DeclareDocumentCommand{\cleardoubleevenpagerplainpage}{}{}
43 \DeclareDocumentCommand{\cleardoubleevenpagerstandardpage}{}{}
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}{\LWR@origmbox{text-align:right}}{}{\dictum}
65   #2
66   \IfValueT{#1}
67   {
68     \ifbool{FormatWP}
69     {\begin{BlockClass}[\LWR@origmbox{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@septestext{#1}%
87 \list{}{}%
88 \let\makelabel\labelinglabel%
89 }
90 {
91 \endlist
92 }
93
94 \DeclareDocumentCommand{\labelinglabel}{m}{%
95 #1 \quad \sc@septestext%
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 \setlength{\LWR@templengthtwo}{#3}
106 \ifblank{#2}

```

```

107 {
108   \begin{BlockClass}[
109     \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthtwo}} ;
110     \LWR@origmbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
111   ]{addmargin}
112 }
113 {
114   \setlength{\LWR@templengthone}{#2}
115   \begin{BlockClass}[
116     \LWR@origmbox{margin-left:\LWR@printlength{\LWR@templengthone}} ;
117     \LWR@origmbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
118   ]{addmargin}
119 }
120 }
121 {\end{BlockClass}}

```

Ref to create a starred environment:

<https://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>

```

122
123 \ExplSyntaxOn
124 \cs_new:cpn {addmargin*} {\addmargin*}
125 \cs_new_eq:cN {endaddmargin*} \endaddmargin
126 \ExplSyntaxOff
127
128 \DeclareDocumentCommand{\marginline}{m}{\marginpar{#1}}

```

File 177 **lwarp-scrhack.sty**

§ 263 Package **scrhack**

Pkg scrhack scrhack is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{scrhack}

File 178 **lwarp-sclayer.sty**

§ 264 Package **sclayer**

(Emulates or patches code by MARKUS KOHM.)

Pkg sclayer sclayer is emulated.

⚠ Not fully tested! Please send bug reports!

```

for HTML output: 1 \LWR@ProvidesPackageDrop{scllayer}

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*{\scrlayerInitInterface}[1] [] {}
63 \newcommand{\scrlayerAddToInterface}[3] [] {}
64 \newcommand{\scrlayerAddCsToInterface}[3] [] {}
65 \newcommand{\scrlayerOnAutoRemoveInterface}[2] [] {}

```

File 179 **lwarp-scrlayer-notecolumn.sty**

§ 265 Package **scrlayer-notecolumn**

(Emulates or patches code by MARKUS KOHM.)

Pkg scrlayer-notecolumn scrlayer-notecolumn is emulated.

 **Not fully tested!** [Please send bug reports!](#)

for HTML output:

```

1 \LWR@ProvidesPackageDrop{scrlayer-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 180 **lwarp-scrlayer-scrpage.sty**

§ 266 Package **scrlayer-scrpage**

(Emulates or patches code by MARKUS KOHM.)

Pkg scrlayer-scrpage scrlayer-scrpage is emulated.

 **Not fully tested!** [Please send bug reports!](#)

for HTML output:

```

1 \LWR@ProvidesPackageDrop{scrlayer-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 m m m m m m}{\}
36 \newcommand*{\defpagestyle}[3]{\}
37 \newcommand*{\newpagestyle}[3]{\}
38 \newcommand*{\providepagestyle}[3]{\}
39 \newcommand*{\renewpagestyle}[3]{\}

```

File 181 **lwarp-section.sty**

§ 267 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{\huge}\else
4  \def\chsize{\huge}\def\hdsi{\Huge}
5 \fi
6 \let\ttsize\LARGE
7 \let\ausize\large
8 \let\dasize\large
9 \let\secsi{\Large}
10 \let\subsi{\large}
11 \let\hdpos\raggedright
12 \newcounter{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\nmidot{}
24 \ifx\secindent\undefined
25  \newdimen\secindent
26  \newskip\secpreskp
27  \newskip\secpstskp
28  \newdimen\subindent
29  \newskip\subpreskp
30  \newskip\subpstskp
31  \newskip\parpstskp
32  \newcount\c@hddepth

```

33 \fi

File 182 **lwarp-sectionbreak.sty**§ 268 Package **sectionbreak***(Emulates or patches code by MICHAL HOFTICH.)*

Pkg sectionbreak sectionbreak is patched for use by lwarp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{sectionbreak}

2 \renewcommand\asterism{\HTMLunicode{2042}}
3
4 \renewcommand\pre@sectionbreak{}
5 \renewcommand\post@sectionbreak{}
6
7 \renewcommand\print@sectionbreak[1]{%
8 \begin{center}
9 #1
10 \end{center}
11 }
12
```

File 183 **lwarp-sectsty.sty**§ 269 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 184 **lwarp-setspace.sty**

§ 270 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 185 **lwarp-shadow.sty**

§ 271 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 186 **lwarp-showidx.sty**

§ 272 Package **showidx**

Pkg showidx showidx is ignored.

for HTML output: Discard all options for lwarp-showidx:

```

1 \LWR@ProvidesPackageDrop{showidx}

```

File 187 **lwarp-showkeys.sty**

§ 273 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 188 **lwarp-sidecap.sty**

§ 274 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 189 **lwarp-sidenotes.sty**

§ 275 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:

```

\sidecaption    * [⟨entry⟩] [⟨offset⟩] {⟨text⟩}

2 \RenewDocumentCommand \sidecaption {s o o +m}
3 {
4     \LWR@stoppars
5     \begingroup
6     \captionsetup{style=sidecaption}
7     \IfBooleanTF{#1}
8     { % starred
9       \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
10       \caption*{#4}
11       \end{BlockClass}
12     }
13     { % unstarred
14       \IfNoValueOrEmptyTF{#2}
15       {\def\@sidenotes@sidecaption@tof{#4}}
16       {\def\@sidenotes@sidecaption@tof{#2}}
17       \begin{BlockClass}[border:none ; box-shadow:none]{marginblock}
18       \caption[\@sidenotes@sidecaption@tof]{#4}
19       \end{BlockClass}
20     }
21     \endgroup

```

```

22 \LWR@startpars
23 }

```

Borrowed from the lwarp version of keyfloat:

```

24 \NewDocumentEnvironment{KFLTsidenotes@marginfloat}{0{-1.2ex} m}
25 {% start
26 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{-}{marginblock}%
27 \captionsetup{type=#2}%
28 }
29 {%
30 \endLWR@BlockClassWP%
31 }
32
33 \RenewDocumentEnvironment{marginfigure}{o}
34 {\begin{KFLTsidenotes@marginfloat}{figure}}
35 {\end{KFLTsidenotes@marginfloat}}
36
37 \RenewDocumentEnvironment{margintable}{o}
38 {\begin{KFLTsidenotes@marginfloat}{table}}
39 {\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_ε article.cls source:

```

40 \renewenvironment{figure*}
41     {\@dblfloat{figure}}
42     {\end@dblfloat}
43
44 \renewenvironment{table*}
45     {\@dblfloat{table}}
46     {\end@dblfloat}

```

File 190 **lwarp-siunitx.sty**

§ 276 Package **siunitx**

(Emulates or patches code by JOSEPH WRIGHT.)

Pkg siunitx siunitx is patched for use by lwarp.

fractions Due to pdftolatex limitations, fraction output is replaced by symbol output for per-mode and quotient-mode.

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

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

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
```

\@ensuredmath is not supported inside an \hbox, so it must temporarily be restored to its original. Similar for \mbox. SVG math is created explicitly when necessary, using \LWR@subsingledollar.

```
9
10 \ExplSyntaxOn
11 %
```

Modified to set set HTML \textcolor if not black:

```
12 \cs_undefine:N \__siunitx_print_aux:
13 \cs_new_protected:Npn \__siunitx_print_aux:
14 {
15   \text
16   {
17     \__siunitx_ensure_ltr:n
18     {
19       \color@begingroup
20       \__siunitx_print_color:
21       \__siunitx_font_shape:
22       \__siunitx_font_weight:
23       \use:c
24       {
25         @@ \l__siunitx_print_type_tl _
26         text \l__siunitx_font_family_tl :
27       }
28       \bool_if:NTF \l__siunitx_font_math_mode_bool
29       { \__siunitx_print_math: }
30       {
31         \LWR@findcurrenttextcolor% lwarp
32         \ifdefstring{\LWR@tempcolor}{000000}% lwarp
33         {\__siunitx_print_text:}% lwarp
34         {% lwarp
35           \LWR@textcurrentcolor{% lwarp
```

```

36             \__siunitx_print_text:
37             }% lwarp
38         }% lwarp
39     }
40     \color@endgroup
41 }
42 }
43 }
44
45
46 \cs_undefine:N \__siunitx_set_math_fam:n
47 \cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
48     \int_new:c { c__siunitx_math #1 _int }
49     \group_begin:% lwarp
50     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
51     \LetLtxMacro\mbox\LWR@origmbox% lwarp
52     \hbox_set:Nn \l__siunitx_tmp_box
53     {
54         \ensuremath
55         {
56             \use:c { math #1 }
57             {
58                 \int_gset:cn { c__siunitx_math #1 _int } { \fam }
59             }
60         }
61     }
62     \group_end:% lwarp
63 }
64
65 \cs_undefine:N \__siunitx_combined_output:n
66 \cs_new_protected:Npn \__siunitx_combined_output:n #1 {
67     \group_begin:% lwarp
68     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
69     \LetLtxMacro\mbox\LWR@origmbox% lwarp
70     \bool_if:NTF \l__siunitx_number_parse_bool
71     {
72         \tl_clear:N \l__siunitx_number_out_tl
73         \bool_set_false:N \l__siunitx_number_compound_bool
74         \__siunitx_number_output_parse:n {#1}
75     }
76     {
77         \__siunitx_unit_output_pre_print:

```

For parse-numbers=false, since \ensuredmath was restored to its original:

```

78 %         \__siunitx_print:nn { number } { \ensuremath {#1} }
79         \LWR@subsingledollar{\__siunitx_print:nn { number } { #1 }}% lwarp
80
81 \__siunitx_unit_output_print:

```

```

81    }
82    \group_end:% lwarp
83 }
84 %

```

For quotients, the fraction code is replaced by the symbol code:

```

85 \cs_undefine:N \__siunitx_number_output_quotient_fraction:
86 \cs_new_protected:Npn \__siunitx_number_output_quotient_fraction: {
87   \bool_set_true:N \l__siunitx_number_compound_bool
88   \__siunitx_number_output_quotient_aux_i:
89   \tl_set_eq:NN \l__siunitx_number_out_tl
90     \l__siunitx_number_numerator_tl
91   \tl_put_right:NV \l__siunitx_number_out_tl \l__siunitx_output_quotient_tl
92   \tl_put_right:NV \l__siunitx_number_out_tl
93     \l__siunitx_number_denominator_tl
94   \__siunitx_number_output_single_aux:
95 }
96 %

```

For units, the fraction code is replaced by the symbol code:

```

97 \cs_undefine:N \__siunitx_unit_format_fraction_fraction:
98 \cs_new_protected:Npn \__siunitx_unit_format_fraction_fraction: {
99   \__siunitx_unit_format_fraction_symbol_aux:
100   \int_compare:nNnT { \l__siunitx_unit_denominator_int } > { 1 }
101     {
102       \bool_if:NT \l__siunitx_unit_denominator_bracket_bool
103         {
104           \tl_put_left:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_open_tl
105           \tl_put_right:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_close_tl
106         }
107       }
108   \tl_set_eq:NN \l__siunitx_unit_tl \l__siunitx_unit_numerator_tl
109   \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_per_symbol_tl
110   \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_unit_denominator_tl
111 }
112
113
114
115 \RenewDocumentCommand \num { o m } {
116   \leavevmode
117   \group_begin:% lwarp
118   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
119   \LetLtxMacro\mbox\LWR@origmbox% lwarp
120   \bool_set_false:N \l__siunitx_font_set_bool
121   \IfNoValueF {#1}
122     { \keys_set:nn { siunitx } {#1} }
123   \__siunitx_number_output:n {#2}

```

```

124 \group_end:% lwarp
125 }
126
127 \RenewDocumentCommand \numrange { o m m } {
128 \leavevmode
129 \group_begin:% lwarp
130 \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
131 \LetLtxMacro\mbox\LWR@origmbox% lwarp
132 \bool_set_false:N \l__siunitx_font_set_bool
133 \IfNoValueF {#1}
134 { \keys_set:nn { siunitx } {#1} }
135 \__siunitx_range_numbers:nn {#2} {#3}
136 \group_end:% lwarp
137 }
138
139 \RenewDocumentCommand \ang { o > { \SplitArgument { 2 } { ; } } m } {
140 \group_begin:% lwarp
141 \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
142 \LetLtxMacro\mbox\LWR@origmbox% lwarp
143 \IfNoValueF {#1}
144 { \keys_set:nn { siunitx } {#1} }
145 \__siunitx_angle_output:nnn #2
146 \group_end:% lwarp
147 }
148
149 \RenewDocumentCommand \si { o m } {
150 \leavevmode
151 \group_begin:% lwarp
152 \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
153 \LetLtxMacro\mbox\LWR@origmbox% lwarp
154 \bool_set_false:N \l__siunitx_font_set_bool
155 \IfNoValueTF {#1}
156 { \__siunitx_unit_output:nn {#2} { } }
157 {
158 \keys_set:nn { siunitx } {#1}
159 \__siunitx_unit_output:nn {#2} {#1}
160 }
161 \group_end:% lwarp
162 }
163
164
165 \RenewDocumentCommand{\SIrange}{o m m m}
166 {%
167 \leavevmode
168 \group_begin:% lwarp
169 \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
170 \LetLtxMacro\mbox\LWR@origmbox% lwarp
171 \bool_set_false:N \l__siunitx_font_set_bool
172 \IfNoValueTF {#1}
173 { \__siunitx_range_unit:nnnn {#4} { } {#2} {#3} }

```

```

174     {
175         \keys_set:nn { siunitx } {#1}
176         \__siunitx_range_unit:nnnn {#4} {#1} {#2} {#3}
177     }
178     \group_end:% lwarp
179 }
180
181 \ExplSyntaxOff

```

File 191 **lwarp-soul.sty**

§ 277 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` $\{\langle text \rangle\} \{\langle color \rangle\} \{\langle class \rangle\} \{\langle colorstyle \rangle\} \{\langle FormatWPstyle \rangle\}$

Add colors if not empty:

```

18 \newcommand{\LWR@soulcolor}[5]{%
19 \ifcempty{#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 192 **lwarp-soulpos.sty**

§ 278 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 193 **lwarp-soulutf8.sty**

§ 279 Package **soulutf8**

Pkg soulutf8 soulutf8 is emulated.

lwarp's HTML output naturally supports UTF-8 encoding.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{soulutf8}
```

File 194 **lwarp-stabular.sty**

§ 280 Package **stabular**

(Emulates or patches code by SIGITAS TOLUŠIS.)

Pkg stabular stabular is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{stabular}

Env stabular [$\langle vpos \rangle$] [$\langle colspec \rangle$]

```
2 \newenvironment{stabular}[2][c]
3 {
4 \begin{tabular}[#1]{#2}
5 \renewcommand{\noalign}[1]{ }
6 }
7 {\end{tabular}}
```

Env stabular [$\langle width \rangle$] [$\langle vpos \rangle$] [$\langle colspec \rangle$]

```
8 \NewDocumentEnvironment{stabular*}{m o m}
9 {
10 \begin{tabular}[#2]{#3}
11 \renewcommand{\noalign}[1]{ }
12 }
13 {\end{tabular}}
```

File 195 **lwarp-stfloats.sty**

§ 281 Package **stfloats**

Pkg stfloats stfloats is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{stfloats}

```
2 \newcommand*{\fnbelowfloat}{}
3 \newcommand*{\fnunderfloat}{}
4 \newcommand*{\setbaselinefloat}{}
5 \newcommand*{\setbaselinefixed}{}


```

File 196 **lwarp-subfig.sty**

§ 282 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 lofentry>] [<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}% lwarp

4 \IfValueTF{#2}{%
5   \LWR@setlatestname{#2}%
6 }{%
7   \IfValueTF{#3}{%
8     \LWR@setlatestname{#3}%
9   }{%
10 }%
11 \LWR@stoppars% lwarp
12   \ifundefined{FBsc@max}{}%
13     {\FB@readaux{\let\FBsuboheight\relax}}%
14   \@tempcnta=\@ne
15   \if@minipage
16     \@tempcnta=\z@
17   \else\ifdim \lastskip=\z@ \else
18     \@tempcnta=\tw@
19   \fi\fi
20   \ifmaincaptiontop
21     \sf@top=\sf@nearskip
22     \sf@bottom=\sf@farskip
```

```

23   \else
24       \sf@top=\sf@farskip
25       \sf@bottom=\sf@nearskip
26   \fi
27   \leavevmode

28 %   \setbox\@tempboxa \hbox{#4}%
29 %   \@tempdima=\wd\@tempboxa
30 %   \@ifundefined{FBsc@max}{}%
31 %       {\global\advance\Xhsize-\wd\@tempboxa
32 %       \dimen@=\ht\@tempboxa
33 %       \advance\dimen@\dp\@tempboxa
34 %       \ifdim\dimen@>\FBso@max
35 %           \global\FBso@max\dimen@
36 %       \fi}%

```

Do not use boxes, which interfere with lateximages:

```

37 %   \vtop%
38   \bgroup
39 %   \vbox%
40   \bgroup
41       \ifcase\@tempcnta
42       \@minipagefalse
43       \or
44       \vskip\sf@top
45       \or
46       \ifdim \lastskip=\z@ \else
47       \@tempskipb\sf@top\relax\@xaddvskip
48       \fi
49   \fi
50   \sf@ifpositiontop{%
51       \ifx \@empty#3\relax \else
52       \sf@subcaption{#1}{#2}{#3}%
53       \vskip\sf@capskip
54       \vskip\sf@captopadj
55   \fi\egroup
56       \hrule width0pt height0pt depth0pt
57       \LWR@startpars% lwarp
58 %   \box\@tempboxa
59 %       #4
60 %       \LWR@stoppars% lwarp
61 %   }{%
62 %       \LWR@startpars% lwarp
63 %       \@ifundefined{FBsc@max}%
64 %       {
65 %   \box\@tempboxa
66 %       #4
67 %       }%
68 %       {\ifx\FBsuboheight\relax

```

```

69 %           \box\@tempboxa
70           #4
71           \else
72 %           \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
73           #4
74           \fi}%
75 \LWR@stoppars% lwarp
76 \egroup
77 \ifx \@empty#3\relax \else
78 %           \vskip\sf@capskip
79 %           \hrule width0pt height0pt depth0pt
80 %           \sf@subcaption{#1}{#2}{#3}%
81           \fi
82       }%
83 %       \vskip\sf@bottom
84 \egroup
85 \@ifundefined{FBsc@max}{}%
86   {\addtocounter{FRobj}{-1}}%
87   \ifnum\c@FRobj=0\else
88     \subfloatrowsep
89   \fi}%
90 \ifmaincaptiontop\else
91   \global\advance\@nameuse{c@\@captype}\m@ne
92 \fi
93 \end{minipage}% lwarp
94 \LWR@startpars% lwarp
95 \endgroup\ignorespaces%
96 }%

```

\sf@subcaption {<1 type>}{<2 lof entry>}{<3 caption>}

```

97 \long\def\sf@subcaption#1#2#3{%
98 \LWR@stoppars% lwarp
99 \ifx \relax#2\relax \else
100   \bgroup
101   \let\label=\@gobble
102   \let\protect=\string
103   \def\@subcaplabel{%
104     \caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
105     \sf@updatecaptionlist{#1}{#2}{\the\value{\@captype}}{\the\value{#1}}%
106   \egroup
107 \fi
108 \bgroup
109 \ifx \relax#3\relax
110   \let\captionlabelsep=\relax
111 \fi
112 %   \setbox0\vbox{%
113 %     \hb@xt@\the\@tempdima{%
114 %

```

```

115 % %          \hss
116 % %          \parbox[t]{\the\@tempdima}{%
117 % %          \caption@make
118 % %          {\@nameuse{sub\@capttype name}}}%
119 % %          {\@nameuse{thesub\@capttype}}}%
120 % %          {#3}
121 % % }%
122 % %          \hss
123 % % }
124 % % }%
125 % \ifundefined{FBsc@max}%
126 %     {\box0}%
127 %     {
128 % \parbox[t]{\the\@tempdima}{%
129 \LWR@traceinfo{sfsubcap B1}% lwarp
130     \LWR@figcaption% lwarp
131     \caption@make
132     {\@nameuse{sub\@capttype name}}}%
133     {\@nameuse{thesub\@capttype}}}%
134     {#3}

135         \endLWR@figcaption% lwarp
136 \LWR@traceinfo{sfsubcap B2}% lwarp
137 % }%
138 % }%
139 % {\dimen@ \ht0%
140 % \advance \dimen@ \dp0%
141 % \ifdim \dimen@ > \FBsc@max
142 %     \global \FBsc@max \dimen@
143 % \fi
144 % \FB@readaux{\let \FBsubcheight \relax}%
145 % \ifx \FBsubcheight \relax
146 %     \def \next{
147 % \parbox[t]{\the\@tempdima}
148 %     }%
149 %     \else
150 %     \def \next{
151 % \parbox[t][\FBsubcheight][t]{\the\@tempdima}
152 %     }%
153 % \fi
154 %     \vbox{%
155 %         \hb@xt@\the\@tempdima{%
156 %
157 %             \hss
158 %             \next{%
159 \LWR@traceinfo{sfsubcap C1}% lwarp
160         \caption@make
161         {\@nameuse{sub\@capttype name}}}%
162         {\@nameuse{thesub\@capttype}}}%
163         {#3}

```

```

164 \LWR@traceinfo{sfsubcap C1}% lwarp
165 % }%
166 %          \hss
167
168 % }
169 %      }
170 %}%
171 \egroup
172 \LWR@startpars% lwarp
173 }

```

\subfloat@label Patches for \sf@sub@label:

```

174 \def\subfloat@label{%
175 \LWR@ensuredoingapar% lwarp
176 \@ifnextchar(% %) match left parenthesis
177   {\sf@sub@label}
178   {\sf@sub@label(Sub\@capytype\space
179                  \@ifundefined{thechapter}{\@nameuse{thechapter}\space}%
180                  \@nameuse{p@sub\@capytype}%
181                  \@nameuse{thesub\@capytype}.)}}

```

Patches for \subref.

\sf@subref {\<label>}

The unstarred version uses a \ref link whose printed text comes from the sub@<label>:

```

182 \renewcommand{\sf@subref}[1]{%
183 \LWR@subnewref{#1}{sub@#1}%
184 }

```

\sf@@subref {\<label>}

The starred version uses the printed sub@<label> which is stored as if it were a page number:

```

185 \renewcommand{\sf@@subref}[1]{\LWR@origpageref{sub@#1}}

```

Defining new subfloats. The l@sub<type> for each is redefined.

\@newsubfloat [{\<keys/values>}] {\<float name>}

```

186 \LetLtxMacro\LWR@orig@newsubfloat\@newsubfloat
187
188 \def\@newsubfloat[#1]#2{%
189 \LWR@orig@newsubfloat[#1]{#2}%
190 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}%
191 }

```

Pre-defined for figures and tables:

```
\l@subfigure  {\langle text\rangle} {\langle pagenum\rangle}
192 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}

\l@subtable  {\langle text\rangle} {\langle pagenum\rangle}
193 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
```

File 197 **lwarp-subfigure.sty**

§ 283 Package **subfigure**

Pkg subfigure subfigure is emulated by subfig.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{subfigure}
2 \RequirePackage{subfig}

3 \LetLtxMacro\subfigure\subfloat
4 \LetLtxMacro\subtable\subfloat
5 \LetLtxMacro\Subref\subref
6 \@ifundefined{figuretopcaptrue}{\newif\iffiguretopcap}{\}
7 \newif\iffiguretopcap
8 \newif\ifsubcaphang
9 \newif\ifsubcapcenter
10 \newif\ifsubcapcenterlast
11 \newif\ifsubcapnooneline
12 \newif\ifsubcapraggedright
13 \newskip\subfigtopskip
14 \newskip\subfigcapskip
15 \newdimen\subfigcaptopadj
16 \newskip\subfigbottomskip
17 \newdimen\subfigcapmargin
18 \newskip\subfiglabelskip
19 \newcommand*{\subcapsize}{}
20 \newcommand*{\subcaplabelfont}{}
21 \newcommand*{\subcapfont}{}

```

File 198 **lwarp-supertabular.sty**

§ 284 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 8.8.

⚠ lateximage supertabular and xtab are not supported inside a lateximage.

```
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   \LWRST@traceinfo{supertabular}%
31   \table%
32   \LWRST@caption%
33   \begin{tabular}{#3}%
34   \TabularMacro\ifdefinedvoid{\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 199 **lwarp-syntonly.sty**

§ 285 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 200 **lwarp-tabls.sty**

§ 286 Package **tabls**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg `tabls` `tabls` is emulated. `\LWR@hline` is used to handle the optional argument when `tabls` is loaded.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{tabls}

2 \newdimen\tablinesep
3 \newdimen\arraylinesep
4 \newdimen\extrarulessep
```

File 201 **lwarp-tabularx.sty**

§ 287 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 \DeclareDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \DeclareDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
```

File 202 **lwarp-tabulary.sty**

§ 288 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 203 **lwarp-textarea.sty**

§ 289 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 204 **lwarp-textcomp.sty**

§ 290 Package **textcomp**

(Emulates or patches code by FRANK MITTELBACH, ROBIN FAIRBAIRNS, WERNER LEMBERG.)

Pkg textcomp textcomp is patched for use by lwarp.

§ 290.1 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.

§ 290.2 Package loading

for HTML output: `1 \LWR@ProvidesPackagePass{textcomp}`

§ 290.3 Remembering original definitions

The following are restored for print when inside a lateximage:

```

2 \let\LWR@origtextdegree\textdegree
3 \let\LWR@origtextcelsius\textcelsius
4 \let\LWR@origtextohm\textohm
5 \let\LWR@origtextmu\textmu
6 \let\LWR@origtextlquill\textlquill
7 \let\LWR@origtexttrquill\texttrquill
8 \let\LWR@origtextcircledP\textcircledP
9 \let\LWR@origtexttwelveudash\texttwelveudash
10 \let\LWR@origtextthrequartersemdash\textthrequartersemdash
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@origcapitallogonek\capitallogonek
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

```

§ 290.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for Lua \TeX and Xe \TeX .

§ 290.4.1 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

```

§ 290.4.2 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{\℧}
65 \renewcommand*\textnaira{\₦}
66 \renewcommand*\textpeso{\₱}

```

```

67 \renewcommand*{\textrecipe}{R}
68 \renewcommand*{\textinterrobangdown}{\_}
69 \renewcommand*{\textpertenthousand}{\%..}
70 \renewcommand*{\textbaht}{\฿}
71 \renewcommand*{\textdiscount}{\}%
72 \renewcommand*{\textservicemark}{\}
73 \fi

```

§ 290.5 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

```

§ 290.6 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\texttrquill\LWR@origtexttrquill%
98 \let\textcircledP\LWR@origtextcircledP%
99 \let\texttwelvewdash\LWR@origtexttwelvewdash%
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 205 **lwarp-textfit.sty**

§ 291 Package **textfit**

Pkg textfit textfit is emulated.

Text is placed into a of class textfit. Sizes are approximated, and also limited by browser min/max font-size settings.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textfit}

2 \newsavebox{\LWR@textfitbox}
3
4 \newcommand*{\LWR@textfitscale}[2]{%
5 \setlength{\LWR@templengthone}{#1}%
6 \setlength{\LWR@templengthone}{%
7   1em*\ratio{\LWR@templengthone}{\LWR@templengthtwo}%
8 }%
9 \InlineClass[font-size:\LWR@printlength{\LWR@templengthone}]{textfit}{#2}%
10 }
11
12 \newcommand*{\scaletowidth}[2]{%
13 \sbox{\LWR@textfitbox}{#2}%
14 \settowidth{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%

```

```

15 \LWR@textfitscale{#1}{#2}%
16 }
17
18 \newcommand*\scaletoheight[2]{%
19 \sbox{\LWR@textfitbox}{#2}%
20 \settoheight{\LWR@templengthtwo}{\usebox{\LWR@textfitbox}}%
21 \LWR@textfitscale{#1}{#2}%
22 }

```

File 206 **lwarp-textpos.sty**

§ 292 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 207 **lwarp-theorem.sty**

§ 293 Package **theorem**

(Emulates or patches code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by lwarp.

for HTML output:

```

1 \LWR@ProvidesPackagePass{theorem}

```

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.

§ 293.1 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     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
8   }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
12  }%
13  \begingroup
14    \csname th@the\theorem@style \endcsname
15  \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
16 \gdef\xnthm#1#2[#3]{%
17   \expandafter\ifdefinable\csname #1\endcsname
18   {%
19     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
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}}}% lwarp
31   }}
32
33 \gdef\@ynthm#1#2{%
34   \expandafter\ifdefinable\csname #1\endcsname
35   {
36     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
37     \definecounter{#1}%
38     \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39     \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40     \expandafter{\csname th@\the \theoremstyle \expandafter
41       \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42     \global \expandafter \let \csname end#1\endcsname \@endtheorem
43     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
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}% lwarp
53       \expandafter \xdef \csname the#1\endcsname
54       {\expandafter \noexpand \csname the#2\endcsname}%
55       \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56       \expandafter{\csname th@\the \theoremstyle \expandafter
57         \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58       \global \expandafter \let \csname end#1\endcsname \@endtheorem
59       \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
60     }%
61   \fi}

```

§ 293.2 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{%
63   \def\@begintheorem##1##2{%
64     \item[
65       \InlineClass{theoremheader}{##1\ ##2}
66     ]

```

```

67    }%
68 \def\@opargbegintheorem##1##2##3{%
69   \item[
70     \InlineClass{theoremheader}{##1\ ##2\ (##3)}
71   ]
72   }
73 }
74
75 \gdef\th@break{%
76   \def\@begintheorem##1##2{%
77     \item[
78       \InlineClass{theoremheader}{##1\ ##2}\newline%
79     ]
80   }%
81 \def\@opargbegintheorem##1##2##3{%
82   \item[
83     \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
84   ]
85   }
86 }
87
88 \gdef\th@marginbreak{%
89   \def\@begintheorem##1##2{
90     \item[
91       \InlineClass{theoremheader}{##2 \quad ##1}\newline
92     ]
93   }%
94 \def\@opargbegintheorem##1##2##3{%
95   \item[
96     \InlineClass{theoremheader}{##2 \quad ##1\ %
97       (##3)}\newline
98   ]
99   }
100 }
101
102 \gdef\th@changebreak{%
103   \def\@begintheorem##1##2{
104     \item[
105       \InlineClass{theoremheader}{##2\ ##1}\newline
106     ]
107   }%
108 \def\@opargbegintheorem##1##2##3{%
109   \item[
110     \InlineClass{theoremheader}{ ##2\ ##1\ %
111       (##3)}\newline
112   ]
113   }
114 }
115
116 \gdef\th@change{%

```

```

117 \def\@begintheorem##1##2{
118   \item[
119     \InlineClass{theoremheader}{##2\ ##1}
120   ]
121   }%
122 \def\@opargbegintheorem##1##2##3{%
123   \item[
124     \InlineClass{theoremheader}{##2\ ##1\ (##3)}
125   ]
126   }
127 }
128
129 \gdef\th@margin{%
130   \def\@begintheorem##1##2{
131     \item[
132       \InlineClass{theoremheader}{##2 \quad ##1}
133     ]
134     }%
135   \def\@opargbegintheorem##1##2##3{%
136     \item[
137       \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
138     ]
139     }
140 }

```

Patched for CSS:

```

141 \gdef\@thm#1#2{\refstepcounter{#1}%
142 \LWR@forcenewpage% lwarp
143   \BlockClass{theorembody\LWR@thisthmstyle}% lwarp
144   \trivlist
145   \@topsep \theorempreskipamount           % used by first \item
146   \@topsepadd \theorempostskipamount        % used by \@endparenv
147   \@ifnextchar [%
148   {\@ythm{#1}{#2}}%
149   {\@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}}
150
151 \gdef\@endtheorem{%
152 \endtrivlist
153 \endBlockClass
154 }

```

File 208 **lwarp-threeparttable.sty**

§ 294 Package **threeparttable**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg	threeparttable	threeparttable is emulated. Table note are contained inside a css <div> of class tnotes. If enumitem is used, the note item labels are also individually highlighted with an additional css of class tnoteitemheader, otherwise they are plain text.
	for HTML output:	1 \LWR@ProvidesPackageDrop{threeparttable}
	\LWR@printtablenote	{\langle text \rangle} Prints the table note item header inside a css class of tnoteitemheader. 2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}
Env	threeparttable	[\langle alignment \rangle] To emulate threeparttable: 3 \newenvironment*{threeparttable}[1][b]{\b}{}
Env	tablenotes	[\langle options \rangle] 4 \newenvironment*{tablenotes}[1][5 {% 6 \LWR@forcenewpage 7 \BlockClass{tnotes}% 8 \ltx@ifpackageloaded{enumitem}{% 9 \setlist[description]{format=\LWR@printtablenote}% 10 }{}% 11 \description% 12 } 13 {% 14 \enddescription% 15 \endBlockClass% 16 } \tnote {\langle text \rangle} 17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}

File 209 **lwarp-tikz.sty**

§ 295 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>

```
2 \newboolean{LWR@tikzbabel}
3
4 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
5 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
6 {\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.

```
7 \BeforeBeginEnvironment{tikzpicture}{%
8 \begin{lateximage}%
9 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
10 {}%
11 {\catcode'\$=3} % dollar sign is math shift
12 }
13
14 \AfterEndEnvironment{tikzpicture}{%
15 \end{lateximage}%
16 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
17 {}%
18 {\catcode'\$=\active}%
19 }
```

File 210 **lwarp-titles.sty**

§ 296 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 {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
2 \NewDocumentCommand{\newpagestyle}{m o m}{}

\renewpagestyle {\langle name \rangle} [\langle style \rangle] {\langle commands \rangle}
3 \NewDocumentCommand{\renewpagestyle}{m o m}{}

\sethead [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}
4 \NewDocumentCommand{\sethead}{o o o m m m}{}

\setfoot [\langle el \rangle] [\langle ec \rangle] [\langle er \rangle] {\langle ol \rangle} {\langle oc \rangle} {\langle or \rangle}
5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

\settitlemarks * {\langle names \rangle}
6 \NewDocumentCommand{\settitlemarks}{s m}{}

\headrule
7 \newcommand*{\headrule}{}

\footrule
8 \newcommand*{\footrule}{}

\setheadrule {\langle length \rangle}
9 \newcommand*{\setheadrule}[1]{}

\setfootrule {\langle length \rangle}
10 \newcommand*{\setfootrule}[1]{}

\makeheadrule
11 \newcommand*{\makeheadrule}{}

<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 <code>\newcommand*{\outertitlemarks}{}{}</code>
<code>\innertitlemarks</code>	
	20 <code>\newcommand*{\innertitlemarks}{}{}</code>
<code>\newtitlemark</code>	<code>* {\langle name \rangle}</code>
	21 <code>\NewDocumentCommand{\newtitlemark}{s m}{}{}</code>
<code>\pretitlemark</code>	<code>* {\langle section \rangle} {\langle text \rangle}</code>
	22 <code>\NewDocumentCommand{\pretitlemark}{s m m}{}{}</code>
<code>\ifsamemark</code>	<code>{\langle group \rangle} {\langle command \rangle} {\langle true \rangle} {\langle false \rangle}</code>
	23 <code>\newcommand{\ifsamemark}[4]{}{}</code>

```

\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 {\<markset>}
31 \newcommand{\topextramarks}[1]{\}

\firstextramarks {\<markset>}
32 \newcommand{\firstextramarks}[1]{\}

\nextextramarks {\<markset>}
33 \newcommand{\nexttopextramarks}[1]{\}

\outerextramarks {\<markset>}
34 \newcommand{\outerextramarks}[1]{\}

\innerextramarks {\<markset>}

```

```
35 \newcommand{\innerextramarks}[1]{}
```

File 211 **lwarp-titleref.sty**

§ 297 Package **titleref**

Pkg titleref titleref is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{titleref}
2
3 \LetLtxMacro\titleref\nameref
4
5 \providecounter{LWR@currenttitle}
6
7 \newcommand*{\currenttitle}{%
8   \addtocounter{LWR@currenttitle}{1}%
9   \label{currenttitle\arabic{LWR@currenttitle}}%
10  \nameref{currenttitle\arabic{LWR@currenttitle}}%
11 }
12
13 \newcommand*{\theTitleReference}[2]{}
```

File 212 **lwarp-titlesec.sty**

§ 298 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    {\label-format}
2 \newcommand*{\titleref}[1]{

\titleref*    {\command} {\format}

\titleref    {\command} [{shape}] {\format} {\label} {\sep} {\before} [{after}]
3 \newcommand\titleref{%
4   \@ifstar{\ttl@format@s}%
```

```

5          {\ttl@format@i}}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}

```

\chaptertitlename

```

8 \@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}

```

```

\titlespacing * {\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}

```

```

\titleline * [\langle align\rangle] {\langle material\rangle}

```

```

19 \NewDocumentCommand{\titleline}{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 213 **lwarp-titletoc.sty**

§ 299 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 m o o} {}
                 5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m m o} {}

```

```

\contentsmargin  [⟨correction⟩] {⟨right⟩}
                 6 \newcommand{\contentsmargin}[2] [] {}

```

\thecontentslabel

```

7 \newcommand*{\thecontentslabel}{thecontentslabel}

\thecontentspage
8 \newcommand*{\thecontentspage}{thecontentspage}

\contentslabel  [⟨format⟩] {⟨space⟩}
9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}

\contentspage  [⟨format⟩]
10 \newcommand{\contentspage}[1] [] {\thecontentspage}

\contentspush  {⟨text⟩}
11 \newcommand{\contentspush}[1] {}

\contentsuse   {⟨name⟩} {⟨text⟩}
12 \newcommand{\contentsuse}[2] {}

\startcontents [⟨name⟩]
13 \newcommand*{\startcontents}[1] [] {}

\stopcontents [⟨name⟩]
14 \newcommand*{\stopcontents}[1] [] {}

\resumecontents [⟨name⟩]
15 \newcommand*{\resumecontents}[1] [] {}

\printcontents [⟨name⟩] {⟨prefix⟩} {⟨start⟩} {⟨code⟩}
16 \newcommand{\printcontents}[4] [] {}

\startlist    [⟨name⟩] {⟨list⟩}
17 \newcommand{\startlist}[2] [] {}

\stoplist     [⟨name⟩] {⟨list⟩}
18 \newcommand{\stoplist}[2] [] {}

```

```

\resumelist  [\langle name \rangle] {\langle list \rangle}
              19 \newcommand{\resumelist}[2] [] {}

\printlist   [\langle name \rangle] {\langle list \rangle} {\langle prefix \rangle} {\langle code \rangle}
              20 \newcommand{\printlist}[4] [] {}


```

File 214 **lwarp-titling.sty**

§ 300 Package **titling**

(Emulates or patches code by PETER WILSON.)

Pkg **titling**

package support lwarp supports the native \TeX 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 57.8.

The various titling footnote restyling commands have no effect.

Pass all options to lwarp-titling:

for HTML output: 1 \LWR@ProvidesPackagePass{titling}

\bsmttitleempty Patch \bsmttitleempty:

```

2 \let\LWR@orig@bsmttitleempty\bsmttitleempty
3 \renewcommand*{\bsmttitleempty}{%
4 \LWR@orig@bsmttitleempty%
5 }

```

\keepthetitle Patch \keepthetitle:

```

6 \let\LWR@origkeepthetitle\keepthetitle
7 \renewcommand*{\keepthetitle}{%
8 \LWR@orig@keepthetitle%
9 }

```

\killtitle Patch \killtitle:

```

10 \let\LWR@origkilltitle\killtitle
11 \renewcommand*{\killtitle}{%
12 \LWR@orig@killtitle%

```

```
13 }
```

Env **titlingpage**

```
14 \renewenvironment*{titlingpage}
15 {%
```

Start an HTML titlepage div:

```
16 \LWR@printpendingfootnotes
17 \begin{titlepage}
```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```
18 \LWR@maketitlesetup
19 \let\maketitle\LWR@titlingmaketitle
20 }
21 {
```

At the end of the environment, end the HTML titlepage div:

```
22 \end{titlepage}
23 }
```

Patch the pre/post title/author/date to add HTML tags, then initialize:

```
24
25 \pretitle{}
26 \posttitle{}
27
28 \preauthor{}
29 \postauthor{}
30
31 \predate{}
32 \postdate{}
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
33 \renewcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
34 \def\@makefnmark{\textsuperscript{\@thefnmark}}
```

`\thefootnote` \Rightarrow `\nameuse{arabic}{footnote}`, or
`\thefootnote` \Rightarrow `\nameuse{fnsymbol}{footnote}`

Redefine the footnote text:

```
35 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
36 \makethanksmark~%
```

```
\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
37 ##1%
38 }% \@makefntext
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     \@bsprefix \@title \@bsprefix
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     \@bsprefixauthor \@author \@bsprefixauthor
65     \end{BlockClass}
66     \end{BlockClass}
67   }
68   \maketitlehookc
69   {
70     \begin{BlockClass}{titledate}
71     \@bsprefixdate \@date \@bsprefixdate
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 [57.7](#).

```
87 \@ifclassloaded{memoir}{
88   \thanksmarkseries{arabic}
89 }{% not memoir
90 \if@titlepage
91   \thanksmarkseries{arabic}
92 \else
93   \thanksmarkseries{fnsymbol}
94 \fi
95 }% not memoir
```

File 215 **lwarp-tocbasic.sty**

§ 301 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}{-}
4 \DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{-}
5 \DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{-}
6 \DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m}{-}
7 \DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m}{-}
8 \DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m}{-}
9 \DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m}{-}
10 \DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m}{-}
11 \DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{-}
12 \DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{-}
13 \DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{-}

```


File 216 **lwarp-tocbibind.sty**

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

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

Inline, with a manual TOC entry:

A commonly-used method to introduce an index in a \LaTeX document:

```

\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex

```

On its own HTML page, with a manual TOC entry:

```

\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex

```

Inline, with an automatic TOC entry:

Pkg tocibind The tocibind package may be used to automatically place an entry in the TOC.

```

\usepackage[nottoc]{tocibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex

```

On its own HTML page, with an automatic TOC entry:

```

\usepackage[nottoc]{tocibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex

```

Opt tocibind numindex Use the tocibind 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{tocibind}

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 \DeclareDocumentCommand{\simplechapter}{0{\@empty}}{%
34     \def\@chapcntformat##1{%
35         #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
36     }%
37 }
38
39 \DeclareDocumentCommand{\restorechapter}{-}{\%
40 \let\@chapcntformat\@secntformat%
41 }


```

File 217 **lwarp-tocloft.sty**

§ 303 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}{}  
  
9 \newcommand*{\cftlofttitlefont}{}  
  
10 \newcommand*{\cftafterlofttitle}{}  
  
11 \newlength{\cftbeforelofttitleskip}  
12 \newlength{\cftafterlofttitleskip}
```

```
\cftlofttitlefont
```

```
9 \newcommand*{\cftlofttitlefont}{}  
  
10 \newcommand*{\cftafterlofttitle}{}  
  
11 \newlength{\cftbeforelofttitleskip}  
12 \newlength{\cftafterlofttitleskip}
```

```
\cftafterlofttitle
```

```
10 \newcommand*{\cftafterlofttitle}{}  
  
11 \newlength{\cftbeforelofttitleskip}  
12 \newlength{\cftafterlofttitleskip}
```

```
\cftmarklot
```

```
13 \newcommand*{\cftmarklot}{}  
  
14 \newcommand*{\cftlottitlefont}{}  
  
15 \newcommand*{\cftafterlottitle}{}  
  
16 \newlength{\cftbeforelottitleskip}  
17 \newlength{\cftafterlottitleskip}
```

```
\cftlottitlefont
```

```
14 \newcommand*{\cftlottitlefont}{}  
  
15 \newcommand*{\cftafterlottitle}{}  
  
16 \newlength{\cftbeforelottitleskip}  
17 \newlength{\cftafterlottitleskip}
```

```
\cftafterlottitle
```

```
15 \newcommand*{\cftafterlottitle}{}  
  
16 \newlength{\cftbeforelottitleskip}  
17 \newlength{\cftafterlottitleskip}
```

```

\cftdot
18 \providecommand*\cftdot}{.}

\cftdotsep
19 \providecommand*\cftdotsep}{1}

\cftnodots
20 \providecommand*\cftnodots}{5000}

\cftdotfill {<sep>}
21 \providecommand\cftdotfill}[1]{

\cftsetpnumwidth {<length>}
22 \DeclareDocumentCommand\cftsetpnumwidth}{m}{}

\cftsetrmarg {<length>}
23 \DeclareDocumentCommand\cftsetrmarg}{m}{}

\cftpnumalign {<alignment>}
24 \DeclareDocumentCommand\cftpnumalign}{m}{}

25 \LWR@providelength\cftparskip}

```

The part-related items are also provided by memoir:

```

26 \LWR@providelength\cftbeforepartskip}
27 \LWR@providelength\cftpartincent}
28 \LWR@providelength\cftpnumwidth}
29 \providecommand*\cftpfont}{}
30 \providecommand*\cftpapresnum}{}
31 \providecommand*\cftpataftersnum}{}
32 \providecommand*\cftpataftersnumb}{}
33 \providecommand*\cftpattleader}{}
34 \providecommand*\cftpardotsep}{1}
35 \providecommand*\cftpapagefont}{}
36 \providecommand*\cftpatafterpnum}{}

```

memoir uses the full name “chapter” instead of “chap”:

```

37 \LWR@providelength{\cftbeforechapskip}
38 \LWR@providelength{\cftchapindent}
39 \LWR@providelength{\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}{}

```

The following do not appear in memoir:

```

48 \LWR@providelength{\cftbeforesecskip}
49 \LWR@providelength{\cftsecindent}
50 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubsecskip}
60 \LWR@providelength{\cftsubsecindent}
61 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubsubsecskip}
71 \LWR@providelength{\cftsubsubsecindent}
72 \LWR@providelength{\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 \LWR@providelength{\cftbeforeparaskip}
82 \LWR@providelength{\cftparaindent}
83 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubparaskip}
93 \LWR@providelength{\cftsubparaindent}
94 \LWR@providelength{\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 \LWR@providelength{\cftbeforefigskip}
104 \LWR@providelength{\cftfigindent}
105 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubfigskip}
115 \LWR@providelength{\cftsubfigindent}
116 \LWR@providelength{\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 \LWR@providelength{\cftbeforetabskip}
```

```

126 \LWR@providelength{\cfttabindent}
127 \LWR@providelength{\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 \LWR@providelength{\cftbeforesubtabskip}
137 \LWR@providelength{\cftsubtabindent}
138 \LWR@providelength{\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 \DeclareDocumentCommand{\cftsetindents}{m m m}{}

148 \newcommand{\pagenumbersoff}[1]{}
149 \newcommand{\pagenumberson}[1]{}

```

\newlistentry [*<within>*] [*<counter>*] [*<ext>*] [*<level-1>*]

```

150 \DeclareDocumentCommand{\newlistentry}{o m m m}
151 {%
152 \LWR@traceinfo{newlistentry #2 #3 #4}%
153 \IfValueTF{#1}%
154 {%
155     \@ifundefined{c@#2}{%
156         \newcounter{#2}[#1]%
157         \expandafter\edef\csname the#2\endcsname{%
158             \expandafter\noexpand\csname the#1\endcsname.\noexpand\arabic{#2}%
159         }%
160     }{}%
161 }%
162 {%
163     \@ifundefined{c@#2}{%
164         \newcounter{#2}%
165     }{}%
166 }%
167 \@namedef{1@#2}##1##2{%

```

```

168 \hypertocfloat{1}{#2}{#3}{##1}{##2}%
169 \def\cftwhatismyname{#2}% from memoir
170}%
171 \expandafter\newlength\csname cftbefore#2skip\endcsname%
172 \expandafter\newlength\csname cft#2indent\endcsname%
173 \expandafter\newlength\csname cft#2numwidth\endcsname%
174 \@namedef{cft#2font}{}%
175 \@namedef{cft#2presnum}{}%
176 \@namedef{cft#2aftersnum}{}%
177 \@namedef{cft#2aftersnumb}{}%
178 \@namedef{cft#2leader}{}%
179 \@namedef{cft#2dotsep}{1}%
180 \@namedef{cft#2pagefont}{}%
181 \@namedef{cft#2afterpnum}{}%
182 \@namedef{toclevel@#2}{#4}%
183 \@namedef{cft#2fillnum}##1{}%
184 \LWR@traceinfo{newlistentry done}%
185 }

```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]
 Emulated through the `\newfloat` mechanism.

```

186 \DeclareDocumentCommand{\newlistof}{o m m m}
187 {%
188 \IfValueTF{#1}
189 {\newlistentry[#1]{#2}{#3}{0}}
190 {\newlistentry{#2}{#3}{0}}
191 \@namedef{ext@#2}{#3}
192 \@ifundefined{c@#3depth}{\newcounter{#3depth}}{}
193 \setcounter{#3depth}{1}
194 \@namedef{cftmark#3}{}
195 \@namedef{listof#2}{\listof{#2}{#4}}
196 \@namedef{cftmake#3title}{}
197 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
198 \expandafter\newlength\csname cftafter#3titleskip\endcsname
199 \@namedef{cft#3titlefont}{}
200 \@namedef{cftafter#3title}{}
201 \@namedef{cft#3prehook}{}
202 \@namedef{cft#3posthook}{}
203 }

```

`\cftchapterprecis` [*<text>*]

```

204 \newcommand{\cftchapterprecis}[1]{%
205 \cftchapterprecishere{#1}
206 \cftchapterprecistoc{#1}}
207 \newcommand{\cftchapterprecishere}[1]{%
208 \begin{quote}\textit{#1}\end{quote}}
209 \newcommand{\cftchapterprecistoc}[1]{

```

```

210 \addtocontents{toc}{%
211   {
212     \protect\begin{quote}#1\protect\end{quote}}
213   }
214 }

```

File 218 **lwarp-tocstyle.sty**

§ 304 Package **tocstyle**

Pkg **tocstyle** **tocstyle** is ignored.

 **Not fully 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 219 **lwarp-todo.sty**

§ 305 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{todolbl:\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 220 **lwarp-todonotes.sty**

§ 306 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}{l@do}{#1}{#2}}

8 \let\LWRTODONOTES@orig@todototoc\todototoc
9
10 \renewcommand*\{\todototoc}{%
11 \phantomsection%
12 \LWRTODONOTES@orig@todototoc%
13 }
14
15 \renewcommand{\@todonotes@drawMarginNoteWithLine}{%
16 \fcolorbox
17   {\@todonotes@currentbordercolor}
18   {\@todonotes@currentbackgroundcolor}
19   {\arabic{\@todonotes@numberoftodonotes}}
20 \marginpar{\@todonotes@drawMarginNote}
21 }
22
23 \renewcommand{\@todonotes@drawInlineNote}{%
24 \fcolorboxBlock%
25   {\@todonotes@currentbordercolor}%
26   {\@todonotes@currentbackgroundcolor}%
27   {%
28     \if@todonotes@authorgiven%
29     {\@todonotes@author:\,%}
30     \fi%
31     \@todonotes@text%
32   }%
33 }
34
35 \renewcommand{\@todonotes@drawMarginNote}{%
36   \if@todonotes@authorgiven%
37     \@todonotes@author\par%
38   \fi%
39   \arabic{\@todonotes@numberoftodonotes}: %
40   \fcolorbox%
41     {\@todonotes@currentbordercolor}%
42     {\@todonotes@currentbackgroundcolor}%
43     {%
44       \@todonotes@sizecommand%
45       \@todonotes@text %
46     }%
47 }%
48
49 \renewcommand{\@todonotes@drawLineToRightMargin}{%}
50
51 \renewcommand{\@todonotes@drawLineToLeftMargin}{%}
52
53 \renewcommand{\missingfigure}[2] [] {%

```

```

54 \setkeys{todonotes}{#1}%
55 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
56 \fcolorboxBlock%
57   {\@todonotes@currentbordercolor}%
58   {\@todonotes@currentfigcolor}%
59   {%
60     \setlength{\fboxrule}{4pt}%
61     \fcolorbox{red}{white}{Missing figure} \quad #2%
62   }
63 }
64
65 \LetLtxMacro\LWRTODONOTES@orig@todo\@todo
66
67 \RenewDocumentCommand{\@todo}{o m}{%
68 \begingroup%
69 \renewcommand*\phantomsection{}%
70 \IfValueTF{#1}{%
71   \LWRTODONOTES@orig@todo[#1]{#2}%
72 }{%
73   \LWRTODONOTES@orig@todo{#2}%
74 }
75 \endgroup%
76 }
77
78 \fi% \if@todonotes@disabled

```

File 221 **lwarp-transparent.sty**

§ 307 Package **transparent**

(Emulates or patches code by HEIKO OBERDIEK.)

Pkg transparent Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

 **Not X_YL_AT_EX!** Note that transparent does not work with X_YL_AT_EX.

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 222 **lwarp-trivfloat.sty**

§ 308 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}
```

§ 308.1 **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 223 **lwarp-turnthepage.sty**

\$ 309 Package **turnthepage**

Pkg **turnthepage** turnthepage is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{turnthepage}

2 \newcommand{\turnthepage}{}

```

File 224 **lwarp-typearea.sty**

\$ 310 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{\recalc\typearea}{}{}
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 225 **lwarp-ulem.sty**

§ 311 Package **ulem**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg ulem Emulated.

for HTML output: Emulate the original package:

```
1 \ProvidesPackage{lwarp-ulem}
```

Original lwarp definitions:

```

2 \LetLtxMacro\LWR@ulemorigemph\emph
3 \LetLtxMacro\LWR@ulemorigtextbf\textbf

```

Basic markup commands, using css:

```

4 \NewDocumentCommand{\uline}{+m}{%
5 \LWR@HTMLtextstyle%
6   {text-decoration:underline;text-decoration-skip}%
7   {uline}{#1}%
8 }
9
10 \NewDocumentCommand{\uuline}{+m}{%
11 \LWR@HTMLtextstyle%
12   {%
13     text-decoration:underline;text-decoration-skip;%
14     text-decoration-style:double%
15   }%
16   {uuline}{#1}%
17 }
18
19 \NewDocumentCommand{\uwave}{+m}{%
20 \LWR@HTMLtextstyle%
21   {%
22     text-decoration:underline;text-decoration-skip;%
23     text-decoration-style:wavy%
24   }%
25   {uwave}{#1}%

```

```

26 }
27
28 \NewDocumentCommand{\sout}{+m}{%
29 \LWR@HTMLtextstyle%
30   {text-decoration:line-through}%
31   {sout}{#1}%
32 }
33
34 \NewDocumentCommand{\xout}{+m}{%
35 \LWR@HTMLtextstyle%
36   {text-decoration:line-through}%
37   {xout}{#1}%
38 }
39
40 \NewDocumentCommand{\dashuline}{+m}{%
41 \LWR@HTMLtextstyle%
42   {%
43     text-decoration:underline;%
44     text-decoration-skip;%
45     text-decoration-style:dashed%
46   }%
47   {dashuline}{#1}%
48 }
49
50 \NewDocumentCommand{\dotuline}{+m}{%
51 \LWR@HTMLtextstyle%
52   {%
53     text-decoration:underline;%
54     text-decoration-skip;%
55     text-decoration-style:dotted%
56   }%
57   {dotuline}{#1}%
58 }

```

Nullified parameters:

```

59 \NewDocumentCommand{\ULthickness}{}{}
60 \newlength{\ULdepth}

```

Nullified/emulated macros:

```

61 \NewDocumentCommand{\markoverwith}{m}{}
62 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}

```

\useunder only works with \textbf, etc, but not \bfseries, etc.

```

63 \NewDocumentCommand{\useunder}{m m m}{%
64 \relax%
65 \ifx\relax#3\relax\else % argumentative command

```

```
66 \def#3{#1}\MakeRobust{#3}\fi
67 }
```

Triggered by package options, also available for the users:

```
68 \newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
69 \newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
70 \ULforem% default
```

Package options:

```
71 \DeclareOption{normalem}{\normalem}
72 \DeclareOption{ULforem}{\ULforem}
73 \DeclareOption{normalbf}{\bf}
74 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}\textbf}
75
76 \DeclareOption*{}
77 \ProcessOptions\relax% original LaTeX code
```

File 226 **lwarp-upref.sty**

§ 312 Package **upref**

Pkg upref Ignored.

for HTML output: Discard all options for lwarp-upref:

```
1 \LWR@ProvidesPackageDrop{upref}
```

File 227 **lwarp-verse.sty**

§ 313 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}
```

<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 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.</p>
--	---

 **spacing** Horizontal spacing relies on pdftotext’s ability to discern the layout (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

Env **verse** The verse environment will be placed inside a HTML `<pre>`.

```
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%
```

<p>Pkg verse</p> <p>Cls memoir</p> <p>\flagverse</p> <p>Len \leftskip</p>	<p>The verse or memoir packages can place stanza numbers to the left with their <code>\flagverse</code> command. Do not allow them to go into the left margin, which would cause pdfcrop to crop the entire page further to the left:</p> <pre>7 \ifdef{\vleftskip}{% 8 \setlength{\vleftskip}{\HTMLvleftskip} 9 \setlength{\leftmargini}{\HTMLleftmargini} 10 }{}</pre>
---	--

```

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 228 **lwarp-vertbars.sty**

§ 314 Package **vertbars**

(Emulates or patches code by PETER WILSON.)

Pkg vertbars vertbars is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{vertbars}

2 \newlength{\barwidth}
3 \setlength{\barwidth}{0.4pt}
4 \newlength{\barspace}
5 \setlength{\barspace}{1em}
6
7 \newenvironment{vertbar}{
8   \LWR@forcenewpage
9   \LWR@forceminwidth{\barwidth}
10  \begin{BlockClass}[%
11    border-left: \LWR@printlength{\LWR@atleastonept} solid black ; %
12    padding-left: \LWR@printlength{\barspace}%

```

```

13   ]{vertbar}
14 }{
15   \end{BlockClass}
16 }

```

File 229 **lwarp-vmargin.sty**

§ 315 Package **vmargin**

Pkg vmargin vmargin is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{vmargin}

```

2 \newcommand*{\LWRVM@customsize}[2]{}
3 \newcommand*{\setpapersize}[2][\ifstrequal{#2}{custom}{\LWRVM@customsize}{}]}
4 \newcommand*{\setmargins}[8]{}
5 \newcommand*{\setmarginsrb}[8]{}
6 \newcommand*{\setmargnohf}[4]{}
7 \newcommand*{\setmargnohfrb}[4]{}
8 \newcommand*{\setmarg}[4]{}
9 \newcommand*{\setmargrb}[4]{}
10 \newlength{\PaperWidth}
11 \setlength{\PaperWidth}{8.5in}
12 \newlength{\PaperHeight}
13 \setlength{\PaperHeight}{11in}
14 \newif\ifLandscape

```

File 230 **lwarp-vwcol.sty**

§ 316 Package **vwcol**

(Emulates or patches code by WILL ROBERTSON.)

Pkg vwcol vwcol is patched for use with lwarp.

The width option is ignored. All vwcol environments adjust to 1–3 equal-width columns, depending on the width of the browser window.

The remaining options are supported, except for lines and maxrecursion.

for HTML output: 1 \LWR@ProvidesPackagePass{vwcol}

Factored from \vwcol. Each is given a style tag to append to the final style.

```

\LWR@vwcol@addrule  {\langle style tag \rangle}

2 \newcommand*{\LWR@vwcol@addrule}[1]{%
3   \appto{\LWR@vwcolstyle}{%
4     #1: %
5     \LWR@printlength{\vwcol@rule} solid \#\LWR@vwcol@rulecolor ; %
6   }%
7 }

\LWR@vwcol@addrule  {\langle style tag \rangle}

8 \newcommand*{\LWR@vwcol@addgap}[1]{%
9   \appto{\LWR@vwcolstyle}{%
10    #1: %
11    \LWR@printlength{\vwcol@sep} ; %
12  }%
13 }

Env   vwcol  {\langle key/values \rangle}

Redefine the environment to add a HTML style. The style is built depending on the
required options.

14 \renewenvironment*{vwcol}[1] [] {%

New paragraph, and process the options:

15 \par\noindent%
16 \vwcolsetup{#1}%

Begin with no style:

17 \newcommand*{\LWR@vwcolstyle}{}

presep and postsep are created with HTML margins:

18 \if@vwcol@presep
19   \appto{\LWR@vwcolstyle}{margin-left: 1em ; padding-left: .5em ; }
20 \fi
21 \if@vwcol@postsep
22   \appto{\LWR@vwcolstyle}{margin-right: 1em ; padding-right: .5em ; }
23 \fi

sep becomes column-gap:

24 \ifdimgreater{\vwcol@sep}{1sp}{
25   \LWR@vwcol@addgap{column-gap}
26   \LWR@vwcol@addgap{-moz-column-gap}
27   \LWR@vwcol@addgap{-webkit-column-gap}
28 }{}

rule become column-rule, while prerule and postrule become HTML borders:

29 \convertcolorspec{named}{\vwcol@rulecol}{HTML}\LWR@vwcol@rulecolor%

```

```

30 \ifdimgreater{\vwcol@rule}{0pt}{
31   \ifdimless{\vwcol@rule}{1pt}{
32     \setlength{\vwcol@rule}{1pt}
33   }{}
34   \LWR@vwcol@addrule{column-rule}
35   \LWR@vwcol@addrule{-moz-column-rule}
36   \LWR@vwcol@addrule{-webkit-column-rule}
37   \if@vwcol@prerule\LWR@vwcol@addrule{border-left}\fi
38   \if@vwcol@postrule\LWR@vwcol@addrule{border-right}\fi
39 }{}

```

Each of the justify options becomes a text-align. Indentation is added where appropriate.

```

40 \ifdefequal{\vwcol@justify}{\RaggedRight}{
41   \appto{\LWR@vwcolstyle}{text-align: left ; }
42   \ifdimgreater{\vwcol@parindent}{0pt}{
43     \appto{\LWR@vwcolstyle}{%
44       text-indent: \LWR@printlength{\vwcol@parindent} ; %
45     }
46   }{}
47 }{}

48 \ifdefequal{\vwcol@justify}{\RaggedLeft}{
49   \appto{\LWR@vwcolstyle}{text-align: right ; }
50 }{}

51 \ifdefequal{\vwcol@justify}{\Centering}{
52   \appto{\LWR@vwcolstyle}{text-align: center ; }
53 }{}

54 \ifdefequal{\vwcol@justify}{\justifying}{
55   \appto{\LWR@vwcolstyle}{text-align: justify ; }
56   \ifdimgreater{\vwcol@parindent}{0pt}{
57     \appto{\LWR@vwcolstyle}{%
58       text-indent: \LWR@printlength{\vwcol@parindent} ; %
59     }
60   }{}
61 }{}

```

Create the <div> with the assembled style:

```

62 \BlockClass[\LWR@vwcolstyle]{multicols}
63 }

```

When the environment ends:

```

64 {
65 \endBlockClass
66 }

```

§ 317 Package **wallpaper**

```
Pkg    wallpaper    wallpaper is emulated.
```

File 232 lwarp-watermark.sty

```
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 233 **lwarp-wrapfig.sty**

§ 319 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 \LWR@BlockClassWP{%
14   width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition; %
15   margin:10pt%
16 }%
17 {%
18   width:\LWR@printlength{\LWR@templengthone}; \LWR@wrapposition%
19 }%
20 {marginblock}%
21 }
22
23
24 \NewDocumentEnvironment{wrapfigure}{o m o m}
25 {%
26 \LWR@subwrapfigure{#2}{#4}%
27 \captionsetup{type=figure}%
28 }
29 {%
30 \endLWR@BlockClassWP%
31 }
32
33
34 \NewDocumentEnvironment{wraptable}{o m o m}
35 {%
36 \LWR@subwrapfigure{#2}{#4}%

```

```

37 \captionsetup{type=table}%
38 }
39 {%
40 \endLWR@BlockClassWP%
41 }
42
43
44 \NewDocumentEnvironment{wrapfloat}{m o m o m}
45 {%
46 \LWR@subwrapfigure{#3}{#5}%
47 \captionsetup{type=#1}%
48 }
49 {%
50 \endLWR@BlockClassWP%
51 }
52
53 \newlength{\wrapoverhang}

```

File 234 **lwarp-xcolor.sty**

§ 320 Package **xcolor**

(Emulates or patches code by DR. UWE KERN.)

Pkg xcolor xcolor is supported by lwarp.

§ 320.1 **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 \rowcolors is supported, except that the optional argument is ignored so far.

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

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

§ 320.2 **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 76. 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 402.

`\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 724.

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 724.

HTML lateximage: Remembers and reuses the print version.

\colorboxBlock:

Print: Becomes \colorbox.

HTML: Newly defined by lwarp-xcolor to use a <div>, page 725.

HTML lateximage: Remembers and reuses the print version \colorbox.

\fcolorbox:

Print: Modified to allow a background of none.

\LWRprint@fcolorbox at section 76

HTML: Redefined by lwarp-xcolor, page 725.

HTML lateximage: Remembers and reuses the print version.

\fcolorboxBlock:

Print: Becomes \fcolorbox. Section 76

HTML: Newly defined by lwarp-xcolor to use a <div>, page 726.

HTML lateximage: Remembers and reuses the print version \fcolorbox.

fcolorminipage:

Print: Newly defined in the lwarp core.

LWRprint@fcolorminipage at section 76

HTML: Newly defined by lwarp-xcolor, page 726.

HTML lateximage: Uses the print version.

\boxframe:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 727.

HTML lateximage: Remembers and reuses the print version.

§ 320.3 Package loading

for HTML output:

```
1 \LWR@ProvidesPackagePass{xcolor}

2 \begin{warpHTML}
```

§ 320.4 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 }
```

§ 320.5 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 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 \convertcolourspec{#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 }

```

§ 320.6 HTML border

`\LWR@borderpadding` `{\colorstyle}\color}` Prints the HTML attributes for a black border and padding.

`\LWR@forceminwidth` must be used first in order to set the border width.

```
56 \newcommand*\LWR@borderpadding[2]{%
57 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
58 padding:\LWR@printlength{\fboxsep}%
59 }
```

§ 320.7 High-level macros

`\color` `\color` appears in the \TeX PDF output, but is ignored by pdftotext and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is `\let` to the final name once the HTML conversion starts.

`\textcolor` `[\model] \color \text`

Converted into an HTML hex color span.

```
60 \RenewDocumentCommand{\textcolor}{0{named} m m}{%
61 \begingroup%
62 \LWR@FBcancel%
63 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
64 \renewcommand*\LWR@currenttextcolor{\#\LWR@tempcolor}%
65 #3%
66 }%
67 \endgroup%
68 }
```

`\pagecolor` `[\model] \color`

Ignored. Use css instead.

```
69 \renewcommand*\pagecolor[2][named]{}
```

`\nopagecolor` Ignored.

```
70 \renewcommand*\nopagecolor{}
```

`\colorbox` `[\model] \color \text`

Converted into an HTML hex background color ``.

```

71 \RenewDocumentCommand{\colorbox}{0{named} m +m}{%
72 \begingroup%
73 \LWR@FBcancel%
74 \InlineClass[%
75 background:\LWR@colorstyle{#1}{#2} ; %
76 padding:\LWR@printlength{\fboxsep}%
77 ]{\colorbox}{#3}%
78 \endgroup%
79 }

```

`\colorboxBlock` [*⟨model⟩*] {*⟨color⟩*} {*⟨text⟩*}

Converted into an HTML hex background color <div>.

```

80 \NewDocumentCommand{\colorboxBlock}{0{named} m +m}{%
81 \begingroup%
82 \LWR@FBcancel%
83 \begin{BlockClass}[%
84 background:\LWR@colorstyle{#1}{#2} ; %
85 padding:\LWR@printlength{\fboxsep}%
86 ]{\colorboxBlock}
87 #3
88 \end{BlockClass}%
89 \endgroup%
90 }

```

`\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.

```

91 \RenewDocumentCommand{\fcolorbox}{0{named} m 0{named} m +m}{%
92 \LWR@traceinfo{HTML fcolorbox #2 #4}%
93 \begingroup%
94 \LWR@FBcancel%
95 \LWR@forceminwidth{\fboxrule}%
96 \ifthenelse{\equal{#4}{none}}{%
97 {% no background color
98   \InlineClass[%
99   \LWR@borderpadding{#1}{#2}%
100 ]{\fcolorbox}{#5}%
101 }%
102 {% yes background color
103   \InlineClass[%
104   \LWR@borderpadding{#1}{#2} ; %
105   background:\LWR@colorstyle{#3}{#4}%

```

```

106    ]{fcolorbox}{#5}%
107 }%
108 \endgroup%
109 }

```

`\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.

```

110 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{named} m +m}{%
111 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
112 \begingroup%
113 \LWR@FBcancel%
114 \LWR@forceminwidth{\fboxrule}%
115 \ifthenelse{\equal{#4}{none}}{%
116 {% no background color
117   \begin{BlockClass}[%
118     \LWR@borderpadding{#1}{#2}%
119   ]{fcolorboxBlock}
120   #5
121   \end{BlockClass}%
122 }%
123 {% yes background color
124   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
125   \begin{BlockClass}[%
126     background:\#\LWR@tempcolortwo\ ; %
127     \LWR@borderpadding{#1}{#2}%
128   ]{fcolorboxBlock}
129   #5
130   \end{BlockClass}%
131 }%
132 \endgroup%
133 \LWR@traceinfo{HTML fcolorboxBlock done}%
134 }

```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the lwarp core: section 76

`\LWR@subfcolorminipage` [*<framemodel>*] [*<framecolor>*] [*<background tag>*] [*<height>*]

```

135 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
136 \begin{BlockClass}[%
137 #3%
138 \LWR@borderpadding{#1}{#2} ; %

```

```

139 \IfValueT{#4}{height:\LWR@printlength{\LWR@tempheight} ; }%
140 width:\LWR@printlength{\LWR@tempwidth}%
141 ]{fcolorminipage}%
142 }

```

Env fcolorminipage [*⟨1:frame model⟩*] [*⟨2:frame color⟩*] [*⟨3:box model⟩*] [*⟨4:box color⟩*] [*⟨5:align⟩*] [*⟨6:height⟩*] [*⟨7:inner-align⟩*] [*⟨8:width⟩*]

```

143 \NewDocumentEnvironment{fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
144 {%
145 \LWR@FBcancel%
146 \setlength{\LWR@tempwidth}{#8}%
147 \IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
148 \LWR@forceminwidth{\fboxrule}%
149 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
150 \ifthenelse{\equal{#4}{none}}%
151 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
152 {%
153     \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
154     \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ;}{#6}%
155 }%
156 }
157 {\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.

```

158 \renewcommand*{\boxframe}[3]{%
159 {%
160 \setlength{\LWR@tempwidth}{#1}%
161 \setlength{\LWR@tempheight}{#2}%
162 \addtolength{\LWR@tempheight}{#3}%
163 \LWR@forceminwidth{\fboxrule}%
164 \InlineClass[%
165 display:inline-block ; %
166 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
167 width:\LWR@printlength{\LWR@tempwidth} ; %
168 height:\LWR@printlength{\LWR@tempheight}%
169 ]{boxframe}{}%
170 }%
171 }

```

§ 320.8 Row colors

```

\rowc@l@rs  [⟨cmds⟩] {⟨startrow⟩} {⟨odd color⟩} {⟨even color⟩}

172 \newcommand*{\LWR@xcolortempcolor}{}
173
174 \def\rowc@l@rs[#1]#2#3#4%
175 {
176 \global\rownum=1
177 \global\@rowcolorstrue
178 \@ifxempty{#3}%
179   {\def\@oddrowcolor{\@norowcolor}}%
180   {%
181     \convertcolorspec{named}{#3}{HTML}\LWR@xcolortempcolor%
182     \edef\@oddrowcolor{%
183       \csdef{LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
184     }%
185   }%
186 \@ifxempty{#4}%
187   {\def\@evenrowcolor{\@norowcolor}}%
188   {%
189     \convertcolorspec{named}{#4}{HTML}\LWR@xcolortempcolor%
190     \edef\@evenrowcolor{%
191       \csdef{LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
192     }%
193   }%
194 \if@rowcmd
195   \def\@rowcolors
196   {%
197 %     #1%
198     \if@rowcolors
199 %       \noalign{%
200 %         \relax\ifnum\rownum<#2\@norowcolor\else
201 %           \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi\fi%
202 %       }%
203     \fi%
204   }%
205 \else
206   \def\@rowcolors
207   {%
208     \if@rowcolors
209 %       \ifnum\rownum<#2%
210 %       \noalign{%
211 %         \@norowcolor
212 %       }
213 %     \else
214 %     #1%
215 %     \noalign{%
216 %       \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi%
217 %     }%

```

```

218          \fi
219          \fi%
220      }%
221      \fi
222      \ignorespaces%
223 }

```

`\@norowcolor` Turns off color for this row.

```

224 \def\@norowcolor{%
225 \renewcommand{\LWR@xcolorrowHTMLcolor}{}}%
226 }

```

`\@rowc@lors` Executed at the end of each row.

```

227 \def\@rowc@lors{%
228 % \noalign{%
229 % \global\advance\rownum\@ne%
230 % }%
231 % \@rowcolors%
232 }

```

```

233 \end{warpHTML}

```

File 235 **lwarp-xfrac.sty**

§ 321 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 \begingroup%
20 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
21 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
22 \endgroup%
23 },
24 denominator-format = {%
25 \begingroup%
```

```

26 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
27 \LWR@htmlsmallfontstart{}\, #1\LWR@htmlsmallfontend%
28 \endgroup%
29 },

```

For pdftotext, do not scale the text:

```

30 scaling = false
31 }
32
33 \DeclareInstance{xfrac}{lmr}{text}{
34 numerator-format = {%
35 \begingroup%
36 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
37 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
38 \endgroup%
39 },
40 denominator-format = {%
41 \begingroup%
42 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
43 \LWR@htmlsmallfontstart{}\, #1\LWR@htmlsmallfontend%
44 \endgroup%
45 },

```

For pdftotext, do not scale the text:

```

46 scaling = false
47 }
48
49 \DeclareInstance{xfrac}{lms}{text}{
50 numerator-format = {%
51 \begingroup%
52 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
53 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
54 \endgroup%
55 },
56 denominator-format = {%
57 \begingroup%
58 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
59 \LWR@htmlsmallfontstart{}\, #1\LWR@htmlsmallfontend%
60 \endgroup%
61 },

```

For pdftotext, do not scale the text:

```

62 scaling = false
63 }
64
65 \DeclareInstance{xfrac}{lmtt}{text}{

```

```

66 numerator-format = {%
67 \begingroup%
68 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
69 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
70 \endgroup%
71 },
72 denominator-format = {%
73 \begingroup%
74 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
75 \LWR@htmlsmallfontstart{\,\,#1\LWR@htmlsmallfontend%
76 \endgroup%
77 },

```

For pdftotext, do not scale the text:

```

78 scaling = false
79 }

```

```

80 \end{warpHTML}

```

File 236 **lwarp-xltxtra.sty**

§ 322 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 237 **lwarp-xmpincl.sty**

§ 323 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*{\includexmp}[1]{}
```


File 238 **lwarp-xtab.sty**

§ 324 Package **xtab**

(Emulates or patches code by PETER WILSON.)

Pkg xtab xtab is emulated.


for HTML output: 1 \LWR@ProvidesPackageDrop{xtab}

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

```

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

See section 8.8.

 **lateximage** supertabular and xtab are not supported inside a lateximage.

```

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 239 **lwarp-xurl.sty**

§ 325 Package **xurl**

Pkg xurl xurl is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{xurl}

File 240 **lwarp-zwpgelayout.sty**

§ 326 Package **zwpgelayout**

(Emulates or patches code by ZDENĚK WAGNER.)

Pkg zwpgelayout zwpgelayout is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{zwpgelayout}

```

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#1{\gdef\ZW@oddwarning}
40\def\SetEvenPageMessage#1{\gdef\ZW@evenwarning}
41\def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning
42
43\def\clap#1{#1}
44
45\def\CropFlap{2in}
46\def\CropSpine{1in}
47\def\CropXSpine{1in}
48\def\CropXtrim{.25in}
49\def\CropYtrim{.25in}
50\def\UserWidth{5in}
51\def\UserLeftMargin{1in}
52\def\UserRightMargin{1in}
53\def\UserTopMargin{1in}
54\def\UserBotMargin{1in}
55\def\thePageNumber{#\,\arabic{page}}
56\ifXeTeX
57\def\ifcaseZWdriver{\ifcase2}
58\else
59\def\ifcaseZWdriver{\ifcase1}
60\fi
61\DeclareRobustCommand\ZWifdriver[2]{\}

```

File 241 **lwarp-patch-komascript.sty**

§ 327 Package **patch-komascript**

Pkg lwarp-patch-komascript Patches for komascript classes.

lwarp loads this package when scrbook, scrartcl, or screprt classes are detected.

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

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

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

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

for HTML output: 1 \ProvidesPackage{lwarp-patch-komascript}

typearea is emulated.

2 \RequirePackage{lwarp-typearea}

tocbasic is emulated.

3 \RequirePackage{lwarp-tocbasic}

scrextend patches most of the new macros.

4 \RequirePackage{lwarp-scrextend}

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 242 **lwarp-patch-memoir.sty**


§ 328 Package **patch-memoir**

(Emulates or patches code by PETER WILSON.)

Pkg lwarp-patch-memoir Patches for memoir class.

 **Not fully tested!** [Please send bug reports!](#)

lwarp loads this package when the memoir class is detected.

 **options clash** While emulating memoir, lwarp pre-loads a number of packages (section 328.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading lwarp:

```

\documentclass{memoir}
...
\PassOptionsToPackage{options_list}{package_name}
...
\usepackage{lwarp}
...
\usepackage{package_name}

```

\verbfootnote is not supported.

\newfootnoteseries, etc. are not supported.

lwarp loads pagenote to perform memoir's pagenote functions, but there are minor differences in \pagenotesubhead and related macros.

Poem numbering is not supported.

The verbatim environment does not yet support the memoir enhancements. It is currently recommended to load and use fancyvrb instead.

The memoir glossary system is not yet supported by lwarpmk. The glossaries package may be used instead, but does require the glossary entries be changed from the memoir syntax to the glossaries syntax.

for HTML output: 1 \ProvidesPackage{lwarp-patch-memoir}

§ 328.1 Packages

These are pre-loaded to provide emulation for many of memoir's functions. memoir pretends that abstract, etc. are already loaded, via its "emulated" package mechanism, but lwarp is directly loading the "lwarp-" version of each, which happens to avoid memoir's emulation system.

```

2 \RequirePackage{lwarp-abstract}% req'd
3 \RequirePackage{lwarp-array}% req'd
4 \RequirePackage{lwarp-booktabs}% req'd
5 % \RequirePackage{lwarp-ccaption}% to do
6 \RequirePackage{lwarp-changepage}% req'd
7 \RequirePackage{lwarp-crop}
8 \RequirePackage{lwarp-dcolumn}% req'd
9 \RequirePackage{lwarp-enumerate}% req'd
10 \RequirePackage{lwarp-epigraph}% req'd
11 \RequirePackage{lwarp-fancyvrb}% req'd
12 \RequirePackage{lwarp-footmisc}% req'd
13 \RequirePackage{lwarp-framed}% req'd
14 \RequirePackage{lwarp-hanging}% req'd
15 \DisemulatePackage{moreverb}
16 \RequirePackage{lwarp-moreverb}
17 \RequirePackage{lwarp-mparhack}

```

```

18 \RequirePackage{lwarp-needspace}% req'd
19 \RequirePackage{lwarp-nextpage}% req'd
20 \RequirePackage{lwarp-pagenote}% req'd
21 \RequirePackage{lwarp-parskip}
22 \RequirePackage{lwarp-setspace}% req'd
23 \RequirePackage{lwarp-showidx}
24 \RequirePackage{lwarp-subfigure}% req'd

```

subfigure is emulated via subfig, which pre-defines subfigure and subtable, but memoir does not, so they must be tested for here:

```

25 \LetLtxMacro\LWR@memorignewsfloat\nnewsfloat
26 \RenewDocumentCommand{\newsfloat}{0{} m}{%
27   \ifundefined{c@sub#2}{%
28     \LWR@memorignewsfloat[#1]{#2}%
29   }{}%
30 }
31
32 \RequirePackage{lwarp-tabularx}% req'd
33 \RequirePackage{lwarp-titling}% req'd
34 % \RequirePackage{lwarp-tocbibind}% not emulated by memoir
35 \RequirePackage{lwarp-tocloft}% req'd
36 \RequirePackage{lwarp-verse}% req'd

```

§ 328.2 Preliminary setup

Bypass the memoir package mechanism:

```

37 \LetLtxMacro\LWR@origlabel\@mem@old@label

```

Redefined to write the LWR@autoindex counter instead of page

```

38 \AtBeginDocument{
39   \def\@wrindexhyp#1||\{\%
40     \addtocounter{LWR@autoindex}{1}%
41     \LWR@newlabel{LWRindex-\arabic{LWR@autoindex}}}%
42   \ifshowindexmark\@showidx{#1}\fi
43   \protected@write\@auxout{%
44     {\string\@wrindexm\@idxfile}{#1}{\arabic{LWR@autoindex}}}%
45   \endgroup
46   \@esphack}%
47 }

```

memoir already set the page size to a default, so it must be forced large for lwarp's use, to avoid tag overflows off the page.

```

48 \setstocksize{190in}{20in}
49 \setlrmarginsandblock{2in}{2in}{*}
50 \setulmarginsandblock{1in}{1in}{*}

```

§ 328.3 Laying out the page

```

51 \renewcommand*{\stockavi}{}
52 \renewcommand*{\stockav}{}
53 \renewcommand*{\stockaiv}{}
54 \renewcommand*{\stockaiii}{}
55 \renewcommand*{\stockbvi}{}
56 \renewcommand*{\stockbv}{}
57 \renewcommand*{\stockbiv}{}
58 \renewcommand*{\stockbiii}{}
59 % \renewcommand*{\stockmetriccrownvo}{}% in docs but not in the package
60 \renewcommand*{\stockmlargecrownvo}{}
61 \renewcommand*{\stockmdemyvo}{}
62 \renewcommand*{\stockmsmallroyalvo}{}
63 \renewcommand*{\pageavi}{}
64 \renewcommand*{\pageav}{}
65 \renewcommand*{\pageaiv}{}
66 \renewcommand*{\pageaiii}{}
67 \renewcommand*{\pagebvi}{}
68 \renewcommand*{\pagebv}{}
69 \renewcommand*{\pagebiv}{}
70 \renewcommand*{\pagebiii}{}
71 % \renewcommand*{\pagemetriccrownvo}{}% in docs but not in the package
72 \renewcommand*{\pagemlargecrownvo}{}
73 \renewcommand*{\pagemdemyvo}{}
74 \renewcommand*{\pagemsmallroyalvo}{}
75
76 \renewcommand*{\stockdbill}{}
77 \renewcommand*{\stockstatement}{}
78 \renewcommand*{\stockexecutive}{}
79 \renewcommand*{\stockletter}{}
80 \renewcommand*{\stockold}{}
81 \renewcommand*{\stocklegal}{}
82 \renewcommand*{\stockledger}{}
83 \renewcommand*{\stockbroadsheet}{}
84 \renewcommand*{\pagedbill}{}
85 \renewcommand*{\pagestatement}{}
86 \renewcommand*{\pageexecutive}{}
87 \renewcommand*{\pageletter}{}
88 \renewcommand*{\pageold}{}
89 \renewcommand*{\pagelegal}{}
90 \renewcommand*{\pageledger}{}
91 \renewcommand*{\pagebroadsheet}{}
92
93 \renewcommand*{\stockpottvo}{}
94 \renewcommand*{\stockfoolscapvo}{}
95 \renewcommand*{\stockcrownvo}{}
96 \renewcommand*{\stockpostvo}{}
97 \renewcommand*{\stocklargecrownvo}{}
98 \renewcommand*{\stocklargepostvo}{}

```

```
99 \renewcommand*{\stocksmallldemyvo}{}
100 \renewcommand*{\stockdemyvo}{}
101 \renewcommand*{\stockmediumvo}{}
102 \renewcommand*{\stocksmallroyalvo}{}
103 \renewcommand*{\stockroyalvo}{}
104 \renewcommand*{\stocksuperroyalvo}{}
105 \renewcommand*{\stockimperialvo}{}
106 \renewcommand*{\pagepottvo}{}
107 \renewcommand*{\pagefoolscapvo}{}
108 \renewcommand*{\pagecrownvo}{}
109 \renewcommand*{\pagepostvo}{}
110 \renewcommand*{\pagelargecrownvo}{}
111 \renewcommand*{\pagelargepostvo}{}
112 \renewcommand*{\pagesmallldemyvo}{}
113 \renewcommand*{\pagedemyvo}{}
114 \renewcommand*{\pagemediumvo}{}
115 \renewcommand*{\pagesmallroyalvo}{}
116 \renewcommand*{\pageroyalvo}{}
117 \renewcommand*{\pagesuperroyalvo}{}
118 \renewcommand*{\pageimperialvo}{}
119
120 \renewcommand*{\memfontfamily}{}
121 \renewcommand*{\memfontenc}{}
122 \renewcommand*{\memfontpack}{}
123
124 \renewcommand*{\anyptfilebase}{}
125 \renewcommand*{\anyptsizesize}{10}
126
127 \renewcommand*{\setstocksize}[2]{}
128 \renewcommand*{\settrimmedsize}[3]{}
129 \renewcommand*{\settrims}[2]{}
130
131 % \newlength{\lxvchars}
132 % \setlength{\lxvchars}{305pt}
133 % \newlength{\xlvchars}
134 % \setlength{\xlvchars}{190pt}
135 \renewcommand*{\setxlvchars}[1]{}
136 \renewcommand*{\setlxvchars}[1]{}
137
138 \renewcommand*{\settypeblocksize}[3]{}
139 \renewcommand*{\setlrmargins}[3]{}
140 \renewcommand*{\setlrmarginsandblock}[3]{}
141 \renewcommand*{\setbinding}[1]{}
142 \renewcommand*{\setulmargins}[3]{}
143 \renewcommand*{\setulmarginsandblock}[3]{}
144 \renewcommand*{\setcolsepandrule}[2]{}
145
146 \renewcommand*{\setheadfoot}[2]{}
147 \renewcommand*{\setheaderspaces}[3]{}
148 \renewcommand*{\setmarginnotes}[3]{}

```

```

149 \renewcommand*{\setfootins}[2]{}
150 \renewcommand*{\checkandfixthelayout}[1] [] {}
151 \renewcommand*{\checkthelayout}[1] {}
152 \renewcommand*{\fixthelayout}{}
153 %
154 % \newlength{\stockheight}
155 % \newlength{\trimtop}
156 % \newlength{\trimedge}
157 % \newlength{\stockwidth}
158 % \newlength{\spinemargin}
159 % \newlength{\foremargin}
160 % \newlength{\uppermargin}
161 % \newlength{\headmargin}
162 %
163 \renewcommand*{\typeoutlayout}{}
164 \renewcommand*{\typeoutstandardlayout}{}
165 \renewcommand*{\settypeoutlayoutunit}[1] {}
166 \renewcommand*{\fixpdflayout}{}
167 \renewcommand*{\fixdvipslayout}{}
168
169 \renewcommand*{\medievalpage}[1] [] {}
170 \renewcommand*{\isopage}[1] [] {}
171 \renewcommand*{\semiisopage}[1] [] {}
172
173 \renewcommand{\setpagebl}[3] {}
174 \renewcommand{\setpageml}[3] {}
175 \renewcommand{\setpagetl}[3] {}
176 \renewcommand{\setpagetm}[3] {}
177 \renewcommand{\setpagetr}[3] {}
178 \renewcommand{\setpagemr}[3] {}
179 \renewcommand{\setpagebr}[3] {}
180 \renewcommand{\setpagebm}[3] {}
181 \renewcommand{\setpagecc}[3] {}

```

§ 328.4 Text and fonts

```

182 \let\miniscule\tiny
183 \let\HUGE\Huge
184
185 \renewcommand*{\abnormalparskip}[1] {}
186 \renewcommand*{\nonzeroparskip}{}
187 \renewcommand*{\traditionalparskip}{}
188
189 \let\onelineskip\baselineskip
190
191 \let\OnehalfSpacing\onehalfspacing
192 \let\DoubleSpacing\doublespacing
193 \renewcommand*{\setPagenoteSpacing}[1] {}
194 \renewcommand*{\setFloatSpacing}[1] {}

```

```

195 \let\SingleSpacing\singlespacing
196 \let\setSingleSpace\SetSinglespace
197 \let\SingleSpace\singlespace
198 \let\endSingleSpace\endsinglespace
199 \let\Spacing\spacing
200 \let\endSpacing\endspacing
201 \let\OnehalfSpace\onehalfspace
202 \let\endOnehalfSpace\endonehalfspace
203 \csletcs{OnehalfSpace*}{onehalfspace}
204 \csletcs{endOnehalfSpace*}{endonehalfspace}
205 \let\DoubleSpace\doublespace
206 \let\endDoubleSpace\enddoublespace
207 \csletcs{DoubleSpace*}{doublespace}
208 \csletcs{endDoubleSpace*}{enddoublespace}
209 \renewcommand*{\setDisplayskipStretch}[1]{
210 \renewcommand*{\memdskipstretch}{
211 \renewcommand*{\noDisplayskipStretch}{
212 \renewcommand*{\memdskips}{
213
214 \renewcommand*{\midsloppy}{
215 \renewenvironment*{\midsloppypar}{
216
217 \renewcommand*{\sloppybottom}{

```

§ 328.5 Titles

```

218 \csletcs{titlingpage*}{titlingpage}
219 \csletcs{endtitlingpage*}{endtitlingpage}
220 \let\andnext\and
221 \renewcommand*{\thanksmarkstyle}[1]{
222 \renewcommand{\thanksfootmark}{%
223 \thanksscript{\tmark}%
224 }
225
226 % \newlength{\thanksmarksep}

```

§ 328.6 Abstracts

```

227 \renewcommand*{\abstractcol}{
228 \renewcommand*{\abstractintoc}{
229 \renewcommand*{\abstractnum}{
230 \renewcommand*{\abstractrunin}{

```

§ 328.7 Document divisions

```

231
232 \def\@apppage{%
233     \part*{\appendixpagename}
234 }

```

```

235 \renewcommand\mempreaddappagetotochook{}
236 \renewcommand\mempostaddappagetotochook{}
237
238 \def\@sappage{%
239     \part*\appendixpagename}
240 }
241
242 \csletcs{frontmatter*}{frontmatter}
243 \csletcs{mainmatter*}{mainmatter}
244 \renewcommand*\raggedbottomsection{}
245 \renewcommand*\normalbottomsection{}
246 \renewcommand*\bottomsectionskip{}
247 \renewcommand*\bottomsectionpenalty{}
248 \csletcs{appendixpage*}{appendixpage}
249 \renewcommand*\namedsubappendices{}
250 \renewcommand*\unnamedsubappendices{}
251 \renewcommand*\setsecnumdepth[1]{}% todo tocvsec2
252 \renewcommand*\maxsecnumdepth[1]{}% todo tocvsec2
253 \renewcommand*\beforebookskip{}
254 \renewcommand*\afterbookskip{}
255 \renewcommand*\beforepartskip{}
256 \renewcommand*\afterpartskip{}
257 \renewcommand*\midbookskip{}
258 \renewcommand*\midpartskip{}
259 \renewcommand*\printbookname{}
260 \renewcommand*\booknamefont{}
261 \renewcommand*\booknamenenum{}
262 \renewcommand*\printbooknum{}
263 \renewcommand*\booknumfont{}
264 \renewcommand*\printpartname{}
265 \renewcommand*\partnamefont{}
266 \renewcommand*\partnamenenum{}
267 \renewcommand*\printpartnum{}
268 \renewcommand*\partnumfont{}
269 \renewcommand*\printbooktitle[1]{}
270 \renewcommand*\booktitlefont{}
271 \renewcommand*\printparttitle[1]{}
272 \renewcommand*\parttitlefont{}
273 \renewcommand*\bookpageend{}
274 \renewcommand*\bookblankpage{}
275 \renewcommand*\nobookblankpage{}
276 \renewcommand*\partpageend{}
277 \renewcommand*\partblankpage{}
278 \renewcommand*\nopartblankpage{}
279 \RenewDocumentCommand{\newleadpage}{s o m m}{}% todo
280 \RenewDocumentCommand{\renewleadpage}{s o m m}{}% todo
281 \renewcommand*\leadpagetoclevel{chapter}
282
283 \renewcommand*\openright{}
284 \renewcommand*\openleft{}

```

```
285 \renewcommand*{\openany}{}
286 \renewcommand*{\clearforchapter}{}
287 \renewcommand*{\memendofchapterhook}{}
288 \renewcommand*{\chapterheadstart}{}
289 % \newlength{\beforechapskip}
290 \renewcommand*{\afterchapternum}{}
291 % \newlength{\midchapskip}
292 \renewcommand*{\afterchaptertitle}{}
293 % \newlength{\afterchapskip}
294 \renewcommand*{\printchaptername}{}
295 \renewcommand*{\chapnamefont}{}
296 \renewcommand*{\chapternamenum}{}
297 \renewcommand*{\printchapternum}{}
298 \renewcommand*{\chapnumfont}{}
299 \renewcommand{\printchaptertitle}[1]{}
300 \renewcommand*{\chaptitelfont}{}
301 \renewcommand*{\printchapternonum}{}
302 \renewcommand*{\indentafterchapter}{}
303 \renewcommand*{\noindentafterchapter}{}
304 \renewcommand*{\insertchapterspace}{}
305
306 \renewcommand*{\chapterstyle}[1]{}
307 \renewcommand{\makechapterstyle}[2]{}
308 \renewcommand*{\chapindent}{}
309 \let\chapterprecis\cftchapterprecis
310 \let\chapterprecishere\cftchapterprecishere
311 \let\chapterprecistoc\cftchapterprecistoc
312 \renewcommand*{\precisfont}{}
313 \renewcommand*{\prechapterprecis}{}
314 \renewcommand*{\postchapterprecis}{}
315 \renewcommand{\precistocetext}[1]{}
316 \renewcommand*{\precistocfont}{}
317 \renewcommand*{\precistocformat}{}
318 % \newlength{\prechapterprecisshift}
319
320 \renewcommand*{\setbeforesecskip}[1]{}
321 \renewcommand*{\setaftersecskip}[1]{}
322 \renewcommand*{\setsecindent}[1]{}
323 \renewcommand*{\setsecheadstyle}[1]{}
324 \renewcommand*{\setbeforesubsecskip}[1]{}
325 \renewcommand*{\setaftersubsecskip}[1]{}
326 \renewcommand*{\setsubsecindent}[1]{}
327 \renewcommand*{\setsubsecheadstyle}[1]{}
328 \renewcommand*{\setbeforesubsubsecskip}[1]{}
329 \renewcommand*{\setaftersubsubsecskip}[1]{}
330 \renewcommand*{\setsubsubsecindent}[1]{}
331 \renewcommand*{\setsubsubsecheadstyle}[1]{}
332 \renewcommand*{\setbeforeparaskip}[1]{}
333 \renewcommand*{\setafterparaskip}[1]{}
334 \renewcommand*{\setparaindent}[1]{}

```

```

335 \renewcommand*\setparaheadstyle}[1]{}
336 \renewcommand*\setbeforesubparaskip}[1]{}
337 \renewcommand*\setaftersubparaskip}[1]{}
338 \renewcommand*\setsubparaindent}[1]{}
339 \renewcommand*\setsubparaheadstyle}[1]{}
340 \renewcommand{\@hangfrom}[1]{#1}
341 \renewcommand{\sethangfrom}[1]{}
342 \renewcommand{\setsecnumformat}[1]{}
343
344 \renewcommand*\hangsecnum{}
345 \renewcommand*\defaultsecnum{}
346
347 \renewcommand*\sechook{}
348 \renewcommand{\setsechook}[1]{}
349 \renewcommand*\subsechook{}
350 \renewcommand{\setsubsechook}[1]{}
351 \renewcommand*\subsubsechook{}
352 \renewcommand{\setsubsubsechook}[1]{}
353 \renewcommand*\parahook{}
354 \renewcommand{\setparahook}[1]{}
355 \renewcommand*\subparahook{}
356 \renewcommand{\setsubparahook}[1]{}
357
358 \RenewDocumentCommand{\plainbreak}{s m}{\begin{center}~\end{center}}
359
360 \RenewDocumentCommand{\fancybreak}{s +m}{%
361 \begin{center}#2\end{center}%
362 }
363
364 \RenewDocumentCommand{\plainfancybreak}{s m m +m}{%
365 \begin{center}#4\end{center}%
366 }
367
368 \RenewDocumentCommand{\pfbreak}{s}{%
369 \begin{center}
370 \pfbreakdisplay
371 \end{center}
372 }
373
374 % \newlength{\pfbreakskip}
375 \renewcommand{\pfbreakdisplay}{*\quad*\quad*}
376
377 \renewcommand{\makeheadstyles}[2]{}
378 \renewcommand*\headstyles[1]{}

```

§ 328.8 Pagination and headers

```

379 \renewcommand*\savepagenumber{}
380 \renewcommand*\restorepagenumber{}

```

```

381 \renewcommand*\uppercaseheads{}\{}
382 \renewcommand*\nouppercaseheads{}\{}
383
384 \renewcommand*\bookpagemark}[1]\{}
385 \renewcommand*\partmark}[1]\{}
386 \renewcommand*\bibmark{}\{}
387 \renewcommand*\indexmark{}\{}
388 \renewcommand*\glossarymark{}\{}
389
390 \LWR@origpagestyle{empty}
391 \renewcommand*\ps@empty{}\{}
392 \renewcommand*\makepagestyle}[1]\{}
393 \renewcommand*\emptyshook{}\{}%
394 % \renewcommand*\empty@oddhead{}\{}
395 % \renewcommand*\empty@oddfoot{}\{}
396 % \renewcommand*\empty@evenhead{}\{}
397 % \renewcommand*\empty@evenfoot{}\{}
398 \renewcommand*\@oddhead{}\{}
399 \renewcommand*\@oddfoot{}\{}
400 \renewcommand*\@evenhead{}\{}
401 \renewcommand*\@evenfoot{}\{}
402 \renewcommand*\aliaspagestyle}[2]\{}
403 \renewcommand*\copypagestyle}[2]\{}
404
405 \renewcommand*\makeevenhead}[4]\{}
406 \renewcommand*\makeoddhead}[4]\{}
407 \renewcommand*\makeevenfoot}[4]\{}
408 \renewcommand*\makeoddfoot}[4]\{}
409 \renewcommand*\makerunningwidth}[3]\{}
410 % \newlength{headwidth}
411 \renewcommand*\makeheadrule}[3]\{}
412 \renewcommand*\makefootrule}[3]\{}
413 \renewcommand*\makeheadfootruleprefix}[3]\{}
414 % \newlength{normalrulethickness}
415 % \setlength{normalrulethickness}{.4pt}
416 % \newlength{footruleheight}
417 % \newlength{footruleskip}
418 \renewcommand*\makeheadposition}[5]\{}
419 \renewcommand*\makepsmarks}[2]\{}
420 \renewcommand*\makeheadfootstrut}[3]\{}
421
422 \renewcommand*\createplainmark}[3]\{}
423 \renewcommand*\memUHead}[1]\{}
424 \renewcommand*\createmark}[5]\{}
425 \renewcommand*\clearplainmark}[1]\{}
426 \renewcommand*\clearmark}[1]\{}
427 \renewcommand*\addtopsmarks}[3]\{}
428 \renewcommand*\ifonlyfloats}[2]{#2}
429 \renewcommand*\mergepagefloatstyle}[3]\{}
430

```

```

431 \renewcommand*\framepichead{}
432 \renewcommand*\framepictextfoot{}
433 \renewcommand*\framepichook{}
434 \renewcommand*\showheadfootlocoff{}
435 \renewcommand*\showtextblocklocoff{}

```

§ 328.9 Paragraphs and lists

```

436 \renewcommand\hangfrom[1]{#1}
437 \let\centerfloat\centering
438 \renewcommand*\raggedyright[1][]{ }
439 % \newlength\ragrparindent
440 \renewcommand\sourceatright[2][]{\attribution{#2}}
441 \let\memorigdb\LWR@endofline
442 \let\memorigpar\par
443 \let\atcentercr\LWR@endofline
444 \renewcommand*\flushleftright{}
445 \renewcommand*\linenottooshort[1][]{ }
446 \renewcommand*\russianpar{}
447 \renewcommand*\lastlinerulefill{}
448 \renewcommand*\lastlineparrule{}
449 \renewcommand*\justlastraggedleft{}
450 \renewcommand*\raggedrightthenleft{}
451 \renewcommand*\leftcenterright{}
452
453 \renewcommand\leftspringright[4]{%
454 \begin{minipage}{#1\linewidth}#3\end{minipage}\quad%
455 \begin{minipage}{#2\linewidth}\begin{flushright}#4\end{flushright}\end{minipage}%
456 }
457
458 \renewenvironment*{blockdescription}
459 {\LWR@descriptionstart\LWR@origdescription}
460 {\enddescription}
461 \renewcommand*\blockdescriptionlabel[1]{\textbf{#1}}
462 \renewenvironment*{labelled}[1]{\begin{description}}{\end{description}}
463 \renewenvironment*{flexlabelled}[6]{\begin{description}}{\end{description}}
464 \renewcommand*\tightlists{}
465 \renewcommand*\defaultlists{}
466 \RenewDocumentCommand\firmlists{s}{}
467 \renewcommand*\firmlist{}
468 \renewcommand*\tightlist{}
469 \renewcommand*\zerotrivseps{}
470 \renewcommand*\savetrivseps{}
471 \renewcommand*\restoretrivseps{}

```

§ 328.10 Contents lists

```

472 \csletcs{tableofcontents*}{tableofcontents}
473 \csletcs{listoffigures*}{listoffigures}

```

```

474 \csletcs{listoftables*}{listoftables}
475 \renewenvironment{KeepFromToc}{}{}
476 \renewcommand*{\onecoltocetc}{}
477 \renewcommand*{\twocoltocetc}{}
478 \renewcommand*{\ensureonecol}{}
479 \renewcommand*{\restorefromonecol}{}
480 \renewcommand*{\doccoltocetc}{}
481 \renewcommand*{\maxtocdepth}[1]{}% tocvsec2
482 \renewcommand*{\settocdepth}[1]{}% tocvsec2
483
484 \renewcommand{\toheadstart}{}
485 \renewcommand{\printtoctitle}[1]{}
486 \renewcommand{\tocmark}{}
487 \renewcommand{\aftertoctitle}{}
488 \renewcommand{\lofheadstart}{}
489 \renewcommand{\printloftitle}[1]{}
490 \renewcommand{\lofmark}{}
491 \renewcommand{\afterloftitle}{}
492 \renewcommand{\lotheadstart}{}
493 \renewcommand{\printlottitle}[1]{}
494 \renewcommand{\lotmark}{}
495 \renewcommand{\afterlottitle}{}
496
497 \renewcommand*{\setpnumwidth}[1]{}
498 \renewcommand*{\setrmarg}[1]{}
499 \renewcommand*{\cftbookbreak}{}
500 \renewcommand*{\cftpartbreak}{}
501 \renewcommand*{\cftchapterbreak}{}

502 % \newlength{\cftbeforebookskip}
503 % \newlength{\cftbookindent}
504 % \newlength{\cftbooknumwidth}
505 \renewcommand*{\cftbookfont}{}
506 \renewcommand*{\cftbookname}{}
507 \renewcommand*{\cftbookpresnum}{}
508 \renewcommand*{\cftbookaftersnum}{}
509 \renewcommand*{\cftbookaftersnumb}{}
510 \renewcommand*{\cftbookleader}{}
511 \renewcommand*{\cftbookdotsep}{1}
512 \renewcommand*{\cftbookpagefont}{}
513 \renewcommand*{\cftbookafterpnum}{}
514 \renewcommand*{\cftbookformatpnum}[1]{}
515 \renewcommand*{\cftbookformatpnumhook}[1]{}

```

Part is already defined by tocloft.

```

516 % \newlength{\cftbeforechapterskip}
517 % \newlength{\cftchapterindent}
518 % \newlength{\cftchapternumwidth}
519 \renewcommand*{\cftchapterfont}{}

```

```

520 \renewcommand*{\cftchaptername}{}
521 \renewcommand*{\cftchapterpresnum}{}
522 \renewcommand*{\cftchapteraftersnum}{}
523 \renewcommand*{\cftchapteraftersnumb}{}
524 \renewcommand*{\cftchapterleader}{}
525 \renewcommand*{\cftchapterdotsep}{1}
526 \renewcommand*{\cftchapterpagefont}{}
527 \renewcommand*{\cftchapterafterpnum}{}
528 \renewcommand*{\cftchapterformatpnum}[1]{}
529 \renewcommand*{\cftchapterformatpnumhook}[1]{}

530 % \newlength{\cftbeforesections skip}
531 % \newlength{\cftsectionindent}
532 % \newlength{\cftsectionnumwidth}
533 \renewcommand*{\cftsectionfont}{}
534 \renewcommand*{\cftsectionname}{}
535 \renewcommand*{\cftsectionpresnum}{}
536 \renewcommand*{\cftsectionaftersnum}{}
537 \renewcommand*{\cftsectionaftersnumb}{}
538 \renewcommand*{\cftsectionleader}{}
539 \renewcommand*{\cftsectiondotsep}{1}
540 \renewcommand*{\cftsectionpagefont}{}
541 \renewcommand*{\cftsectionafterpnum}{}
542 \renewcommand*{\cftsectionformatpnum}[1]{}
543 \renewcommand*{\cftsectionformatpnumhook}[1]{}

544 % \newlength{\cftbeforesubsection skip}
545 % \newlength{\cftsubsectionindent}
546 % \newlength{\cftsubsectionnumwidth}
547 \renewcommand*{\cftsubsectionfont}{}
548 \renewcommand*{\cftsubsectionname}{}
549 \renewcommand*{\cftsubsectionpresnum}{}
550 \renewcommand*{\cftsubsectionaftersnum}{}
551 \renewcommand*{\cftsubsectionaftersnumb}{}
552 \renewcommand*{\cftsubsectionleader}{}
553 \renewcommand*{\cftsubsectiondotsep}{1}
554 \renewcommand*{\cftsubsectionpagefont}{}
555 \renewcommand*{\cftsubsectionafterpnum}{}
556 \renewcommand*{\cftsubsectionformatpnum}[1]{}
557 \renewcommand*{\cftsubsectionformatpnumhook}[1]{}

558 % \newlength{\cftbeforesubsubsection skip}
559 % \newlength{\cftsubsubsectionindent}
560 % \newlength{\cftsubsubsectionnumwidth}
561 \renewcommand*{\cftsubsubsectionfont}{}
562 \renewcommand*{\cftsubsubsectionname}{}
563 \renewcommand*{\cftsubsubsectionpresnum}{}
564 \renewcommand*{\cftsubsubsectionaftersnum}{}
565 \renewcommand*{\cftsubsubsectionaftersnumb}{}
566 \renewcommand*{\cftsubsubsectionleader}{}

```

```
567 \renewcommand*{\cftsubsubsectiondotsep}{1}
568 \renewcommand*{\cftsubsubsectionpagefont}{}
569 \renewcommand*{\cftsubsubsectionafterpnum}{}
570 \renewcommand*{\cftsubsubsectionformatpnum}[1]{}
571 \renewcommand*{\cftsubsubsectionformatpnumhook}[1]{}

572 % \newlength{\cftbeforeparagraphskip}
573 % \newlength{\cftparagraphindent}
574 % \newlength{\cftparagraphnumwidth}
575 \renewcommand*{\cftparagraphfont}{}
576 \renewcommand*{\cftparagraphname}{}
577 \renewcommand*{\cftparagraphpresnum}{}
578 \renewcommand*{\cftparagraphaftersnum}{}
579 \renewcommand*{\cftparagraphaftersnumb}{}
580 \renewcommand*{\cftparagraphleader}{}
581 \renewcommand*{\cftparagraphdotsep}{1}
582 \renewcommand*{\cftparagraphpagefont}{}
583 \renewcommand*{\cftparagraphafterpnum}{}
584 \renewcommand*{\cftparagraphformatpnum}[1]{}
585 \renewcommand*{\cftparagraphformatpnumhook}[1]{}

586 % \newlength{\cftbeforesubparagraphskip}
587 % \newlength{\cftsubparagraphindent}
588 % \newlength{\cftsubparagraphnumwidth}
589 \renewcommand*{\cftsubparagraphfont}{}
590 \renewcommand*{\cftsubparagraphname}{}
591 \renewcommand*{\cftsubparagraphpresnum}{}
592 \renewcommand*{\cftsubparagraphaftersnum}{}
593 \renewcommand*{\cftsubparagraphaftersnumb}{}
594 \renewcommand*{\cftsubparagraphleader}{}
595 \renewcommand*{\cftsubparagraphdotsep}{1}
596 \renewcommand*{\cftsubparagraphpagefont}{}
597 \renewcommand*{\cftsubparagraphafterpnum}{}
598 \renewcommand*{\cftsubparagraphformatpnum}[1]{}
599 \renewcommand*{\cftsubparagraphformatpnumhook}[1]{}

600 % \newlength{\cftbeforefigureskip}
601 % \newlength{\cftfigureindent}
602 % \newlength{\cftfigurenumwidth}
603 \renewcommand*{\cftfigurefont}{}
604 \renewcommand*{\cftfigurename}{}
605 \renewcommand*{\cftfigurepresnum}{}
606 \renewcommand*{\cftfigureaftersnum}{}
607 \renewcommand*{\cftfigureaftersnumb}{}
608 \renewcommand*{\cftfigureleader}{}
609 \renewcommand*{\cftfiguredotsep}{1}
610 \renewcommand*{\cftfigurepagefont}{}
611 \renewcommand*{\cftfigureafterpnum}{}
612 \renewcommand*{\cftfigureformatpnum}[1]{}
613 \renewcommand*{\cftfigureformatpnumhook}[1]{}

```

```

614 % \newlength{\cftbeforesubfigureskip}
615 % \newlength{\cftsubfigureindent}
616 % \newlength{\cftsubfigureenumwidth}
617 \newcommand*{\cftsubfigurefont}{}
618 \newcommand*{\cftsubfigurename}{}
619 \newcommand*{\cftsubfigurepresnum}{}
620 \newcommand*{\cftsubfigureaftersnum}{}
621 \newcommand*{\cftsubfigureaftersnumb}{}
622 \newcommand*{\cftsubfigureleader}{}
623 \newcommand*{\cftsubfiguredotsep}{1}
624 \newcommand*{\cftsubfigurepagefont}{}
625 \newcommand*{\cftsubfigureafterpnum}{}
626 \newcommand*{\cftsubfigureformatpnum}[1]{}
627 \newcommand*{\cftsubfigureformatpnumhook}[1]{}

628 % \newlength{\cftbeforetableskip}
629 % \newlength{\cfttableindent}
630 % \newlength{\cfttableenumwidth}
631 \renewcommand*{\cfttablefont}{}
632 \renewcommand*{\cfttablename}{}
633 \renewcommand*{\cfttablepresnum}{}
634 \renewcommand*{\cfttableaftersnum}{}
635 \renewcommand*{\cfttableaftersnumb}{}
636 \renewcommand*{\cfttableleader}{}
637 \renewcommand*{\cfttabledotsep}{1}
638 \renewcommand*{\cfttablepagefont}{}
639 \renewcommand*{\cfttableafterpnum}{}
640 \renewcommand*{\cfttableformatpnum}[1]{}
641 \renewcommand*{\cfttableformatpnumhook}[1]{}

642 % \newlength{\cftbeforesubtableskip}
643 % \newlength{\cftsubtableindent}
644 % \newlength{\cftsubtableenumwidth}
645 \newcommand*{\cftsubtablefont}{}
646 \newcommand*{\cftsubtablename}{}
647 \newcommand*{\cftsubtablepresnum}{}
648 \newcommand*{\cftsubtableaftersnum}{}
649 \newcommand*{\cftsubtableaftersnumb}{}
650 \newcommand*{\cftsubtableleader}{}
651 \newcommand*{\cftsubtabledotsep}{1}
652 \newcommand*{\cftsubtablepagefont}{}
653 \newcommand*{\cftsubtableafterpnum}{}
654 \newcommand*{\cftsubtableformatpnum}[1]{}
655 \newcommand*{\cftsubtableformatpnumhook}[1]{}

656 \renewcommand*{\booknumberline}[1]{}
657 \renewcommand*{\partnumberline}[1]{}
658 \renewcommand*{\chapternumberline}[1]{}
659 \renewcommand*{\numberlinehook}[1]{}
660 % \renewcommand*{\cftwhatismyname}{}%
```

```

661 \renewcommand*{\booknumberlinehook}[1]{}
662 \renewcommand*{\partnumberlinehook}[1]{}
663 \renewcommand*{\chapternumberlinehook}[1]{}
664 \renewcommand{\numberlinebox}[2]{}
665 \renewcommand{\booknumberlinebox}[2]{}
666 \renewcommand{\partnumberlinebox}[2]{}
667 \renewcommand{\chapternumberlinebox}[2]{}
668 %
669 % \newlength{\cftparskip}
670 \renewcommand*{\cftpagenumbersoff}[1]{}
671 \renewcommand*{\cftpagenumberson}[1]{}
672 \renewcommand*{\cftlocalchange}[3]{}
673 \renewcommand*{\cftaddtitleline}[4]{}
674 \renewcommand*{\cftaddnumtitleline}[4]{}
675 \renewcommand{\cftinsertcode}[2]{}
676 \renewcommand{\cftinserthook}[2]{}
677 \renewcommand{\settocpreprocessor}[2]{}
678 \DeclareRobustCommand{\cftpagenumbersoff}[1]{}
679 \DeclareRobustCommand{\cftpagenumberson}[1]{}

```

§ 328.11 Floats and captions

`\newfloat` [*<1: within>*] [*<2: type>*] [*<3: ext>*] [*<4: capname>*]

```

680 \RenewDocumentCommand{\newfloat}{o m m m}{%
681 \IfValueTF{#1}%
682 {\DeclareFloatingEnvironment[fileext=#3,within=#1,name={#4}]{#2}}%
683 {\DeclareFloatingEnvironment[fileext=#3,name={#4}]{#2}}}%

```

`newfloat` package automatically creates the `\listof` command for new floats, but `float` does not, so remove `\listof` here in case it is manually created later.

```

684 \cslet{listof#2s}\relax%
685 \cslet{listof#2es}\relax%
686 }

```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]

Emulated through the `\newfloat` mechanism. Note that `memoir` uses a different syntax than `tocloft` for the name.

```

687 \RenewDocumentCommand{\newlistof}{o m m m}{
688 {%
689 \IfValueTF{#1}
690 {\newlistentry[#1]{#2}{#3}{0}}
691 {\newlistentry[#2]{#3}{0}}
692 \@namedef{ext@#2}{#3}%
693 \@ifundefined{c@#3depth}{\newcounter{#3depth}}{}%
694 \setcounter{#3depth}{1}%
695 \@namedef{#3mark}{}%
696 \@namedef{#2}{\listof{#2}{#4}}

```

```

697 \@namedef{@cftmake#3title}{}
698 \@ifundefined{cftbefore#3titleskip}{
699     \expandafter\newlength\csname cftbefore#3titleskip\endcsname
700     \expandafter\newlength\csname cftafter#3titleskip\endcsname
701 }{}
702 \@namedef{cft#3titlefont}{}
703 \@namedef{cftafter#3title}{}
704 \@namedef{cft#3prehook}{}
705 \@namedef{cft#3posthook}{}
706 }

```

```

707 \renewcommand{\setfloatadjustment}[2]{}

```

Borrowed from the lwarp version of keyfloat:

```

708 \NewDocumentEnvironment{KFLTmemoir@marginfloat}{0}{-1.2ex}{m}
709 {% start
710 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
711 \captionsetup{type=#2}%
712 }
713 {%
714 \endLWR@BlockClassWP%
715 }
716
717 \DeclareDocumentEnvironment{marginfigure}{o}{}
718 {\begin{KFLTmemoir@marginfloat}{figure}}
719 {\end{KFLTmemoir@marginfloat}}
720
721 \DeclareDocumentEnvironment{margintable}{o}{}
722 {\begin{KFLTmemoir@marginfloat}{table}}
723 {\end{KFLTmemoir@marginfloat}}
724
724 \renewcommand{\setmarginfloatcaptionadjustment}[2]{}
725 \renewcommand{\setmpjustification}[2]{}
726 \renewcommand*{\mpjustification}{}
727 \renewcommand*{\setfloatlocations}[2]{}
728 \DeclareDocumentCommand{\suppressfloats}{o}{}
729 \renewcommand*{\FloatBlock}{}
730 \renewcommand*{\FloatBlockAllowAbove}{}
731 \renewcommand*{\FloatBlockAllowBelow}{}
732 \renewcommand*{\setFloatBlockFor}{}
733 \renewcommand*{\captiondelim}[1]{\renewcommand*{\CaptionSeparator}{#1}}
734 \renewcommand*{\captionnamefont}[1]{}
735 \renewcommand*{\captiontitlefont}[1]{}
736 \renewcommand*{\captionstyle}[2][{}
737 \renewcommand*{\centerlastline}{}
738 \renewcommand*{\hangcaption}{}
739 \renewcommand*{\indentcaption}[1]{}
740 \renewcommand*{\normalcaption}{}

```

```

741 \renewcommand*{\changecaptionwidth}{%
742 \DeclareDocumentCommand{\captionwidth}{m}{%
743 \renewcommand*{\normalcaptionwidth}{%
744 \renewcommand{\precaption}[1]{%
745 \renewcommand{\captiontitlefinal}[1]{%
746 \renewcommand{\postcaption}[1]{%
747
748 \renewcommand{\contcaption}[1]{%
749 % \ContinuedFloat%
750 % \caption{#1}%
751 \begin{LWR@figcaption}% later becomes \caption*
752 \csuse{\@capttype name} \thechapter.\the\value{\@capttype}\CaptionSeparator #1
753 \end{LWR@figcaption}
754 }

```

The extra `\\` here forces a `
` in HTML when `\legend` is used in a `\marginpar`.

```

755 \renewcommand{\legend}[1]{\begin{center}#1\\end{center}}
756
757 \renewcommand{\namedlegend}[2][{}{
758 \begin{center}
759 \csuse{fleg\@capttype}\CaptionSeparator#2\\
760 \end{center}
761 \csuse{flegtoc\@capttype}{#1}
762 }
763
764 \renewcommand{\flegtable}{\tablename}
765 \renewcommand{\flegfigure}{\figurename}
766 \renewcommand{\flegtocfigure}{%
767 \renewcommand{\flegtocfigure}{%
768
769 \renewcommand{\newfixedcaption}[3][\caption]{%
770 \renewcommand{#2}{\def\@capttype{#3}#1}}
771 \renewcommand{\renewfixedcaption}[3][\caption]{%
772 \renewcommand{#2}{\def\@capttype{#3}#1}}
773 \renewcommand{\providefixedcaption}[3][\caption]{%
774 \providecommand{#2}{\def\@capttype{#3}#1}}
775
776 \renewcommand{\bitwonumcaption}[6][{}{
777 \ifblank{#2}{\caption{#3}}{\caption{#2}{#3}}%
778 \addtocounter{\@capttype}{-1}%
779 \begingroup%
780 \csdef{\@capttype name}{#4}%
781 \ifblank{#5}{\caption{#6}}{\caption{#5}{#6}}%
782 \endgroup%
783 \ifblank{#1}{\label{#1}}%
784 }
785
786 \LetLtxMacro\bionenumcaption\bitwonumcaption% todo

```

```

787
788 \renewcommand{\bication}[5][{}]{%
789 \ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
790 \begin{LWR@figcaption}% later becomes \caption*
791 #4 \thechapter.\the\value{\@capttype}\CaptionSeparator #5
792 \end{LWR@figcaption}
793 \ifblank{#1}{\label{#1}}%
794 }
795
796 \renewcommand{\bicontcaption}[3]{%
797 \contcaption{#1}%
798 \begingroup%
799 \csdef{\@capttype name}{#2}%
800 \contcaption{#3}%
801 \endgroup%
802 }
803
804 \renewcommand{\midbication}[1]{%
805
806 \renewcommand{\subcaption}[2][{}]{%
807 \ifblank{#1}{\subfloat[#2]}{\subfloat[#1][#2]}%
808 }
809
810 \RenewDocumentCommand{\subtop}{O{} O{} m}{%
811 \subfloat[#1][#2]{#3}%
812 }
813
814 \RenewDocumentCommand{\subbottom}{O{} O{} m}{%
815 \subfloat[#1][#2]{#3}%
816 }
817
818 \renewcommand{\contsubcaption}{\ContinuedFloat\subcaption}
819
820 \renewcommand{\contsubtop}{%
821 \ContinuedFloat\addtocounter{\@capttype}{1}%
822 \subtop}
823
824 \renewcommand{\contsubbottom}{%
825 \ContinuedFloat\addtocounter{\@capttype}{1}%
826 \subbottom}
827
828 \renewcommand{\subconcluded}{}
829
830 \LetLtxMacro\subcaptionref\subref
831
832 \renewcommand*{\tightsubcaptions}{}
833 \renewcommand*{\loosesubcaptions}{}
834
835 \renewcommand*{\subcaptionsize}[1]{}
836 \renewcommand*{\subcaptionlabelfont}[1]{}

```

```

837 \renewcommand*{\subcaptionfont}[1]{}
838 \renewcommand*{\subcaptionstyle}[1]{}
839
840 \renewcommand*{\hangsubcaption}{}
841 \renewcommand*{\shortsubcaption}{}
842 \renewcommand*{\normalsubcaption}{}
843
844 \RenewDocumentEnvironment{sidecaption}{o m o}
845 {}
846 {
847 \IfValueTF{#1}{\caption[#1]{#2}}{\caption{#2}}%
848 \IfValueT{#3}{\label{#3}}%
849 }
850
851 % \newlength{\sidecapwidth}
852 % \newlength{\sidecapsep}
853 \renewcommand*{\setsidecaps}[2]{}
854 \renewcommand*{\sidecapmargin}[1]{}
855 % \newif\ifscapmargleft
856 \scapmargleftfalse
857 \renewcommand*{\setsidecappos}[1]{}
858
859 \RenewDocumentEnvironment{sidecontcaption}{m o}
860 {}
861 {%
862 \ContinuedFloat%
863 \caption{#1}%

```

Without \@capttype, the section is referred to instead.

```

864 \IfValueT{#2}{\label[\@capttype]{#2}}%
865 }

```

\sidenamedlegend does not appear to use the TOC argument.

```

866 \renewenvironment{sidenamedlegend}[2] [] {
867 \begin{center}
868 \csuse{\@capttype name}\CaptionSeparator#2
869 \end{center}
870 }
871 {}
872
873 \renewenvironment{sidelegend}[1]
874 {\begin{center}
875 #1
876
877 }
878 {\end{center}}
879

```

```

880 \renewcommand*{\sidecapstyle}{}
881 \renewcommand*{\overridescapmargin}[1]{}
882 % \newlength{\sidecapraise}
883 \renewcommand*{\sidecapfloatwidth}{\linewidth}
884
885 \LetLtxMacro\ctabular\ctabular
886 \LetLtxMacro\endctabular\endctabular
887
888 \renewcommand{\autorows}[5][\%
889 #5
890 }
891
892 \renewcommand{\autocol}[5][\%
893 #5
894 }

```

§ 328.12 Page notes

```

895 \renewcommand*{\feetabovelfloat}{}
896 \renewcommand*{\feetbelowfloat}{}
897 \renewcommand*{\feetatbottom}{}
898
899 \renewcommand*{\verbfootnote}[2][\{
900 \PackageError{lwarp, memoir}
901 {Verbatim footnotes are not yet supported by lwarp.}
902 {This may be improved some day.}
903 }
904
905 \renewcommand*{\plainfootnotes}{}
906 \renewcommand*{\twocolumnfootnotes}{}
907 \renewcommand*{\threecolumnfootnotes}{}
908 \renewcommand*{\paragraphfootnotes}{}
909 \renewcommand*{\footfudgefiddle}{}
910
911 \renewcommand*{\newfootnoteseries}[1]{
912 \PackageError{lwarp, memoir}
913 {Memoir footnote series are not yet supported by lwarp.}
914 {This may be improved some day.}
915 }
916
917 \renewcommand*{\plainfootstyle}[1]{}
918 \renewcommand*{\twocolumnfootstyle}[1]{}
919 \renewcommand*{\threecolumnfootstyle}[1]{}
920 \renewcommand*{\paragraphfootstyle}[1]{}
921
922 \renewcommand*{\footfootmark}{}
923 \renewcommand*{\footmarkstyle}[1]{}
924
925 % \newlength{\footmarkwidth}

```

```

926 % \newlength{\footmarksep}
927 % \newlength{\footparindent}
928
929 \renewcommand*{\foottextfont}{}
930
931 \renewcommand*{\marginparmargin}[1]{}
932 \renewcommand*{\sideparmargin}[1]{}
933
934 \LetLtxMacro\sidepar\marginpar
935 \renewcommand*{\sideparfont}{}
936 \renewcommand*{\sideparform}{}
937 \LWR@providelength{\sideparvshift}
938
939 \renewcommand*{\parnopar}{}
940
941 \renewcommand{\sidebar}[1]{\begin{quote}#1\end{quote}}
942 \renewcommand*{\sidebarmargin}[1]{}
943 \renewcommand*{\sidebarfont}{}
944 \renewcommand*{\sidebarform}{}
945 % \newlength{\sidebarhsep}
946 % \newlength{\sidebarvsep}
947 % \newlength{\sidebarwidth}
948 % \newlength{\sidebartopsep}
949 \renewcommand{\setsidebarheight}[1]{}
950 \renewcommand*{\setsidebars}[6]{}
951 \renewcommand*{\footnotesatfoot}{}
952 \renewcommand*{\footnotesinmargin}{}
953
954 \LetLtxMacro\sidefootnote\footnote
955 \LetLtxMacro\sidefootnotemark\footnotemark
956 \LetLtxMacro\sidefootnotetext\footnotetext
957
958 \renewcommand*{\sidefootmargin}[1]{}
959 % \newlength{\sidefoothsep}
960 % \newlength{\sidefootvsep}
961 % \newlength{\sidefootwidth}
962 % \newlength{\sidefootadjust}
963 % \newlength{\sidefootheight}
964 \renewcommand*{\setsidefootheight}[1]{}
965 % \renewcommand*{\sidefootfont}{}% in docs but not in the package
966 \renewcommand*{\setsidefeet}[6]{}
967 \renewcommand*{\sidefootmarkstyle}[1]{}
968 \renewcommand*{\sidefoottextfont}{}
969 \renewcommand*{\sidefootform}{}
970
971 \renewcommand*{\continuousnotenums}{\pncontopttrue}% from pagenote
972 \renewcommand*{\notepageref}{}
973 \renewcommand*{\prenotetext}{}
974 \renewcommand*{\postnotetext}{}
975 \renewcommand*{\idtextinnotes}[1]{}

```

```

976 \renewcommand*{\printpageinnotes}[1]{ }
977 \renewcommand*{\printpageinnoteshyperref}[1]{ }
978 \renewcommand*{\foottopagenote}{ }
979 \renewcommand*{\pagetofootnote}{ }

```

§ 328.13 Decorative text

```

980 \renewcommand*{\epigraphposition}[1]{ }
981 \renewcommand*{\epigraphtextposition}[1]{ }
982 \renewcommand*{\epigraphsourceposition}[1]{ }
983 \renewcommand*{\epigraphfontsize}[1]{ }
984 \renewcommand*{\epigraphforheader}[2][ ]{ }
985 \renewcommand*{\epigraphpicture}{ }

```

§ 328.14 Poetry

```

986 \renewcommand*{\vinphantom}{ }
987 \renewcommand*{\vleftofline}[1]{#1}
988 % \let\linenumberfrequency\poemlines
989 % \renewcommand*{\linenumberfont}[1]{ }
990
991 \DeclareDocumentCommand{\PoemTitle}{s o m}{%
992 \IfValueTF{#2}%
993 {\poemtitle[#2]{#4}}%
994 {\poemtitle{#4}}%
995 }
996
997 \renewcommand*{\NumberPoemTitle}{ }
998 \renewcommand*{\PlainPoemTitle}{ }
999 \renewcommand*{\poemtitlestyle}{ }
1000 \renewcommand*{\poemtitlestarmark}[1]{ }
1001 \renewcommand*{\poemtitlestarpstyle}{ }
1002 \renewcommand*{\PoemTitleheadstart}{ }
1003 \renewcommand*{\printPoemTitlenonum}{ }
1004 \renewcommand*{\printPoemTitlenum}{ }
1005 \renewcommand*{\afterPoemTitlenum}{ }
1006 \renewcommand*{\printPoemTitletitle}[1]{ }
1007 \renewcommand*{\afterPoemTitle}{ }
1008 \newlength{\midpoemtitleskip}
1009 \renewcommand*{\PoemTitlenumfont}{ }
1010 \renewcommand*{\PoemTitlefont}{ }

```

§ 328.15 Boxes, verbatims and files

```

1011 \renewenvironment{qframe}{\framed}{\endframed}
1012 \renewenvironment{qshade}{\shaded}{\endshaded}

```

Use the comment package:

```

1013 \renewcommand*{\commentsoff}[1]{\includecomment{#1}}
1014 \renewcommand*{\commentson}[1]{\excludecomment{#1}}
1015 \LetLtxMacro\renewcomment\commentson
1016
1017 \renewcommand*{\setverbatimfont}[1]{}
1018 \renewcommand*{\tabson}[1]{}
1019 \renewcommand*{\tabsoff}{}
1020 \renewcommand*{\wrappingon}{}
1021 \renewcommand*{\wrappingoff}{}
1022 \renewcommand*{\verbatimindent}{}
1023 \renewcommand*{\verbatimbreakchar}[1]{}

1024 \DefineVerbatimEnvironment{fboxverbatim}{Verbatim}{frame=single}

```

boxedverbatim is already defined by moreverb. boxedverbatim* does not appear to work at all, even in a minimal print memoir document.

```

1025 \renewcommand*{\bvbox}{}
1026 \renewcommand*{\bvtopandtail}{}
1027 \renewcommand*{\bvsiides}{}
1028 \renewcommand*{\nobvbox}{}
1029 % \newlength\bvboxsep
1030 \renewcommand*{\bvtoprulehook}{}
1031 \renewcommand*{\bvtopmidhook}{}
1032 \renewcommand*{\bvendrulehook}{}
1033 \renewcommand*{\bvleftsidehook}{}
1034 \renewcommand*{\bvrightsidehook}{}
1035 \renewcommand*{\bvperpagetrue}{}
1036 \renewcommand*{\bvperpagefalse}{}
1037 \renewcommand*{\bvtopofpage}[1]{}
1038 \renewcommand*{\bvendofpage}[1]{}
1039 \renewcommand*{\linenumberfrequency}[1]{}
1040 \renewcommand*{\resetbvlینumber}{}
1041 \renewcommand*{\setbvlینums}[2]{}
1042 \renewcommand*{\linenumberfont}[1]{}
1043 \renewcommand*{\bvnumbersinside}{}
1044 \renewcommand*{\bvnumbersoutside}{}

```

§ 328.16 Cross referencing

```

1045 \renewcommand*{\fref}[1]{\cref{#1}}
1046 \renewcommand*{\tref}[1]{\cref{#1}}
1047 \renewcommand*{\pref}[1]{\cpageref{#1}}
1048 \renewcommand*{\Aref}[1]{\cref{#1}}
1049 \renewcommand*{\Bref}[1]{\cref{#1}}
1050 \renewcommand*{\Pref}[1]{\cref{#1}}
1051 \renewcommand*{\Sref}[1]{\cref{#1}}
1052 \renewcommand*{\figurerefname}{Figure}
1053 \renewcommand*{\tablerefname}{Table}

```

```

1054 \renewcommand*{\pagerefname}{page}
1055 \renewcommand*{\bookrefname}{Book~}
1056 \renewcommand*{\partrefname}{Part~}
1057 \renewcommand*{\chapterrefname}{Chapter~}
1058 \renewcommand*{\sectionrefname}{\S}
1059 \renewcommand*{\appendixrefname}{Appendix~}
1060 \LetLtxMacro\titleref\nameref
1061 \renewcommand*{\headnameref}{}
1062 \renewcommand*{\tocnameref}{}
1063
1064 \providecounter{LWR@currenttitle}
1065
1066 \renewcommand*{\currenttitle}{%
1067     \addtocounter{LWR@currenttitle}{1}%
1068     \label{currenttitle\arabic{LWR@currenttitle}}%
1069     \nameref{currenttitle\arabic{LWR@currenttitle}}%
1070 }
1071
1072 \renewcommand*{\theTitleReference}[2]{}
1073 \renewcommand*{\namerefon}{}
1074 \renewcommand*{\namerefoff}{}

```

§ 328.17 Back matter

```

1075 \DeclareDocumentCommand{\newblock}{}{}
1076 %
1077 \renewcommand*{\showindexmarks}{}
1078 \renewcommand*{\hideindexmarks}{}
1079
1080 \renewcommand*{\xindyindex}{}

```

§ 328.18 Miscellaneous

```

1081 \renewcommand*{\changemarks}{}
1082 \renewcommand*{\nochangemarks}{}
1083 \renewcommand*{\added}[1]{}
1084 \renewcommand*{\deleted}[1]{}
1085 \renewcommand*{\changed}[1]{}
1086
1087 \renewcommand*{\showtrimsoff}{}
1088 \renewcommand*{\showtrimson}{}
1089 \renewcommand*{\trimXmarks}{}
1090 \renewcommand*{\trimLmarks}{}
1091 \renewcommand*{\trimFrame}{}
1092 \renewcommand*{\trimNone}{}
1093 \renewcommand*{\trimmarkscolor}{}
1094 \renewcommand*{\trimmarks}{}
1095 \renewcommand*{\tmarktl}{}
1096 \renewcommand*{\tmarktr}{}

```

```
1097 \renewcommand*{\tmarkbr}{  
1098 \renewcommand*{\tmarkbl}{  
1099 \renewcommand*{\tmarktm}{  
1100 \renewcommand*{\tmarkmr}{  
1101 \renewcommand*{\tmarkbm}{  
1102 \renewcommand*{\tmarkml}{  
1103 \renewcommand*{\trimmark}{  
1104 \renewcommand*{\quarkmarks}{  
1105 \renewcommand*{\registrationColour}[1]{  
1106  
1107 \renewcommand*{\leavespergathering}[1]{  
1108  
1109 \renewcommand*{\noprelistbreak}{  
1110  
1111 \renewcommand*{\cleartorecto}{  
1112 \renewcommand*{\cleartoverso}{  
1113  
1114 \renewenvironment{vplace}[1][ ]{}{
```

§ 328.19 Final patchwork

```
1115 \newlistof{tableofcontents}{toc}{\contentsname}  
1116 \newlistof{listoffigures}{lof}{\listfigurename}  
1117 \newlistof{listoftables}{lot}{\listtablename}
```

Change History and Index

§ 328 Change History

v0.10	General: 2016/03/08 Initial version . . . 1	Docs: Table: Float data structures. 383
v0.11	General: 2016/03/11 1	Docs: Trademarks section. 474
	Added section: Operating-System	Docs: Troubleshooting
	portability. 143	cross-references. 134
	Added section: Selecting the	Test Suite: Assigned cleveref name
	operating system. 91	for Test Float. 1
	Test Suite: MS-WINDOWS in	Test Suite: Floatrow 1
	README.txt 1	
	Test Suite: limages and index in	
	README.txt 1	
v0.12	\LWR@newhtmlfile: Bugfix: TOC	
	with numbered files. 269	
	General: 2016/03/14 1	
	Global: Uses \p@(type) in float	
	captions. 1	
	Test Suite: Sub-figures 1	
v0.13	\CaptionSeparator: Fix for newer	
	babel package. 386	
	\LWR@LwarpStart: \up and \fup . . 285	
	General: 2016/03/24 1	
	Removed package: subfig 1	
	Test Suite: Ordinals, Subcaption . . 1	
	tikzpicture: Fix dollar-redefined	
	bug for newer package. 681	
v0.14	\LWR@htmlsectionfilename: Fix:	
	Links to home page. 235	
	General: 2016/03/31 1	
	floatrow: Added. 535	
	Docs: Commands for a successful	
	HTML conversion. 96	
	Docs: Commands into a warpprint	
	environment. 93	
	Docs: Newclude limitations. . . . 121	
	Docs: Table: Cross-referencing	
	data structures. 372	
		General: 2016/04/06 1
		Added 540
		Ampersand (&): Fixed handling
		when passed as an argument. . . 322
		Docs: Added warning icons for
		items needing special attention. 140
		Docs: Clarify print/HTML output. 92
		Docs: Moved the supported
		features table to the introduction. 52
		Files: lwarp_formal.css added. . . . 1
		Fix: steps counter 539
		Fixed & handling. 537
		Test Suite: test_suite_formal.css
		file added. 1
		v0.16
		General: 2016/04/11 1
		\titlingpage: Improved
		print-output spacing. 294
		xfrac: Adjusted for the use of any
		font: 730
		Added XeLaTeX, LuaLaTeX
		support. 155
		Docs: Font and UTF-8 support. . . 81
		Docs: Moved location of
		\usepackage{lwarp}. 83
		Docs: Text not converting. 134
		Fix: amsmath options clash 159
		Fix: newtxmath compatibility. . . 159
		Lwarp no longer selects fonts. 81, 155
		Removed package: suffix 1
		Test Suite: Improved titlingpage. 294

Test Suite: Lwarp no longer selects fonts.	1	verse: Supports verse, memoir packages.	712
Test Suite: Supports XeLaTeX, LuaLaTeX.	1	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage.	447
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page.	235	\HTMLFilename: Docs: Escape filename underscores.	234
General: 2016/04/14	1	\HomeHTMLFilename: Docs: Escape filename underscores.	234
mdframed: Added.	592	\LWR@LwarpStart: Enabled \\ equal to \newline.	284
Test Suite: Fix: Print-version front-matter page numbers.	1	\LWR@doubledollar: MATHJAX support.	408
Test Suite: Mdframed	1	\LWR@filestart: lwarp_mathjax.txt loaded.	282
v0.18		\LWR@hspace: Fix: \hspace length computations.	464
\LWR@hspace: \hspace supported.	464	\LWR@minipagestartpars: Surpresses paragraph tags between minipages.	463
\LWR@includegraphicsb: Add: svgz file extension.	556	\LWR@singledollar: MATHJAX support.	409
em, ex, %, px dimensions preserved.	556	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size.	425
Fix: \linewidth, \textwidth, \textheight inside a minipage.	556	\minipagefullwidth: Added: No width tag for the next minipage in HTML.	446
Improved HTML output linebreaks.	556	\rule: Added	468
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic.	390	\warpHTMLonly: Added.	147
\LWR@newhtmlfile: sideroc after title, improving responsive design.	268	\warpprintonly: Replaces \rowprintedonly.	147
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic.	287	\xfracHTMLfontsize: Added.	730
\LWR@subhyperref: Improved HTML output linebreaks.	380	General: 2016/06/08	1
\LWR@subhyperrefclass: Improved HTML output linebreaks.	380	MATHJAX support added.	412, 416, 417
\LWR@subinlineimage: Surpress extra space.	381	multirow: Added optional args.	608
General: 2016/05/19	1	Avoids MATHJAX.	408
File: lwarp.css: Improved TOC outline display.	1	cleveref: Loaded \AtEndPreamble.	442
Files: lwarp.css and lwarp_formal.css: Improved responsive design.	1	CSS for table note item.	680
Microtype disabled during HTML generation	155	Docs: Math options.	83
PDF Unicode input characters.	141	Docs: Table: Cross-referencing data structures, updated.	372
Test Suite: Verse package	1	File: lwarp.css: tnoteitemheader added.	1
lateximage: pdfcrop: --hires added.	429	File: lwarp_mathjax.txt added.	1
Reorganize \HomeHTMLFilename logic.	429	Introduction: MATHJAX support mentioned.	50
Surpress extra space.	429	Options: mathsvg and mathjax	145

Supports colored <code>\rule</code>	723	<code>\LWR@subhtmlclass</code> :	
<code>titles: null \pagestyle</code> and		Factored code.	245
<code>\thispagestyle</code> for HTML. . .	681	<code>\SetHTMLFileNumber</code> : Add: Control	
<code>tikzpicture</code> : Adapts to tikz version.	681	file numbers.	234
<code>equation: MATHJAX</code> support.	414	<code>\cpagerefFor</code> : User-redefinable	
v0.20		word for page references.	443
<code>\BlockClassSingle</code> : Renamed from		<code>\dotfill</code> : Inserts an ellipsis.	462
" <code>LWR@htmldivclassline</code> ".	247	<code>\hfill</code> : Inserts a <code>\quad</code>	462
<code>\HTMLDescription</code> : Added		<code>\hrulefill</code> : Inserts a short rule.	462
<code>\NewHTMLdescription</code> .		<code>\hyperindexref</code> : Print mode	
(Renamed in v0.30.)	257	provided in case <code>hyperref</code> not	
<code>\HTMLFilename</code> : No longer escape		used.	400
<code>underscores</code>	234	<code>\pageref</code> : Added.	379
<code>\HomeHTMLFilename</code> : No longer		<code>\tracinglwrap</code> : Added.	166
escape <code>underscores</code>	234	General: 2017/02/09	1
<code>\InlineClass</code> : Renamed from		<code>afterpage</code> : Added.	482
" <code>inlineclass</code> ".	247	<code>alltt</code> : Added.	484
<code>\LWR@LwrapStart</code> : Fix: math cross		<code>bookmark</code> : Added.	495
references.	286	caption and subcaption supported.	1
<code>\LWR@closeparagraph</code> : <code>\unskip</code>		<code>cleveref</code> and referencing patches:	
extra spaces.	251	Applied <code>\AfterEndPreamble</code>	442
No break tags in the start/end of a		<code>draftwatermark</code> : Added.	509
tabular.	251	<code>eso-pic</code> : Added.	515
<code>\LWR@endoffline</code> : Fix: <code>\\</code>	462	<code>everypage</code> : Added.	516
<code>\LWR@filestart</code> : Adds meta		<code>extramarks</code> : Added.	516
<code>description</code>	282	<code>fancyhdr</code> : Added.	521
<code>\LWR@hspace</code> : Add: Supports HTML		<code>hyperref</code> : Additional user macros.	567
thin breakable space.	464	<code>keyfloat</code> : Added.	574
<code>\LWR@htmldivclass</code> : Added		<code>letterspace</code> : User-interface	
optional style.	245	emulated.	577
<code>\LWR@htmlclass</code> : Added		<code>listings</code> : Added.	581
optional style.	245	<code>ltxcaption</code> : Added.	587
<code>\LWR@htmlsectionfilename</code> :		<code>lwrap-newproject</code> : Added.	175
<code>HTMLFilename</code> : removed		<code>microtype</code> : User-interface	
additional trailing <code>'</code> , and may be		emulated.	603
empty.	235	<code>needspace</code> : Added.	612
Sections called "Index" or "index"		<code>nowidow</code> : Added.	614
have an underscore prepended to		<code>placeins</code> : Added.	631
their filenames if no prefix.	235	<code>ragged2e</code> : Added.	633
<code>\LWR@includegraphicsb</code> : Fix:		<code>setspace</code> : Improved support.	648
<code>\linewidth</code> in a floatrow.	556	<code>textpos</code> : Added.	675
Fix: Expands filename.	556	<code>titles</code> : Added.	681
<code>\LWR@longtablecaptiontag</code> :		<code>titlesec</code> : Added.	685
Fix: Pairs in captions.	355	<code>titletoc</code> : Added.	687
<code>\LWR@section</code> : Combined		<code>titling</code> : Improved compatibility.	689
higher-level sections together		<code>tocloft</code> : Added.	696
into files.	274	<code>wallpaper</code> : Added.	717
<code>\LWR@setOSWindows</code> : Auto-detects		<code>wrapfig</code> : Added.	718
operating system.	143	Added <code>@</code> , <code><</code> , <code>></code> columns.	316

Added single-expansion data arrays.	232	lwarpmk: Fix: lwarpmk limages for WINDOWS.	217
Code factored into independent lwarpmk files.	474	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script.	217
Docs: Examples for generating HTML file names.	89	Add: Errors for misplaced packages.	148
Docs: Improved index.	1	Docs: Added internet class.	56
Enhanced titling support.	292	Docs: Added TeX2page, GladTeX.	56
File: lwarpmk.css: Minor fixes for validation.	1	Docs: Installing on WINDOWS.	62
File: lwarpmk used to compile print, HTML, indexes, and lateximages.	1	File: lwarpmk_tutorial.txt added.	67
Fix: \linewidth in a floatrow.	537	v0.22	
Improved float caption type handling.	532	\LWR@parseDcolumn: Added tabular D column.	329
Moved sidebar and example code to test suite.	1	\LWR@parsebangcolumn: Added tabular ! column.	327
Page geometry set to 6in wide with large margins.	156	\LWR@parsetablecols: Unknown table column types become 1. Added tabular D, !, X columns.	330
Parallel versions of aux files for print/HTML.	1	\LWR@printmccoldata: Added tabular D, !, and X columns.	351
Removed reliance on make, grep, gawk.	1	General: 2017/03/02	1
Tabular: \unskip extra spaces.	316	abstract: Added.	476
Test Suite: HTML meta descriptions.	1	changepage: Added.	502
verbatim: Added.	303	dcolumn: Added.	508
BlockClass: Added optional style.	247	ftnright: Added.	548
Renamed from "blockclass".	247	geometry: Nullified commands.	550
LWR@nestspan: Fix: Minipages inside a span.	242	indentfirst: Added.	573
v0.21		layout: Added.	576
\LWR@LwarpStart: Changed lateximages to a .txt file.	284	lscape: Added.	587
\LWR@filestart: Skip title if not given.	282	mcaption: Added.	591
\LWR@newhtmlfile: Skip title if not given.	268	nameref: Added.	611
\marginpar: Fixed source listing.	263	nextpage: Added.	613
\marginparBlock: Fixed source listing.	263	parskip: Added.	629
General: 2017/02/23	1	showkeys: Added.	650
fontenc: Added.	542	sidecap: Added.	650
fontspec: Added.	543	tabularx: Added.	669
inputenc: Added.	574	varioref: Supported.	103
newclude: Added.	613	verse: Added.	711
newunicodechar: Added.	613	v0.23	
lwarpmk: Fix: lwarpmk again for WINDOWS.	217	\LWR@parsetablecols: Fix for vert bar column type.	330
		\LWR@printmccoldata: Fix for vert bar column type.	351
		General: 2017/03/02	1
		v0.24	
		\LWR@hspace: Add: \hspace \fill converts to 2em	464
		\LWR@htmlfileref: Fix: Index links while \tracinglwarpmk.	375

\hypertocfloat: List of floats responds to lofdepth, lotdepth.	397	mparhack: Added.	606
General: 2017/03/15	1	pagenote: Supported as-is.	628
floatrow: Support for subfig.	535	sidenotes: Added.	651
subfig: Added.	661	Docs: Improved MiKTeX install instructions.	61, 63
tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary.	680	Dollar span avoided in a lateximage.	408
Docs: Filename underscore.	86, 97	Footnotes now are \LaTeX boxes instead of pagenotes.	258
No longer preloads subcaption; conflicted with subfig.	158	lateximage: Labels track page numbers of lateximages.	429
picture: Fix for inline images.	444	Print mode now uses a minipage of \linewidth.	429
tikzpicture: Fix for inline images.	681	picture: Fix for \makebox in picture.	444
v0.25		v0.27	
\LWR@loadnever: Added the ability to prevent conflicting packages.	149	\LWR@footnotetext: Fix for table footnote par tags.	259
\addcontentsline: Handles theorems.	389	General: 2017/04/04	1
General: 2016/03/22	1	lettrine: Added.	577
amsthm: Added.	484	microtype: Fix with Xe \LaTeX , Lua \LaTeX	603
ccaption: Prevented.	501	soul: Added.	657
ellipsis: Added.	510	ulem: Added.	709
emptypage: Added.	511	Docs: Installing utilities for MACOS.	64
framed: Added.	546	Docs: Limitations of saveboxes.	98
lips: Added.	581	Page geometry modified to reduce line overflow.	156
mdframed: Help avoid hyphenation.	594	v0.28	
nththeorem: Added.	615	\@wrindex: Improved indexing.	400
showidx: Added.	649	\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.)	257
theorem: Added.	675	\LWR@LwarpEnd: If FormatEPUB or FormatWP, no bottom nav.	287
Basic \LaTeX theorems: improved css.	304	\LWR@LwarpStart: FormatWordProcessor forces single-file output.	284
Docs: Adds credits for patched code.	1	\LWR@filestart: Adds HTML meta author.	282
Docs: Testing lwarp.	133	\LWR@forcenewpage: Forces new PDF page before major environments.	238
Fix: Allows Xe \LaTeX and Lua \LaTeX to preload graphics and graphicx.	150	\LWR@htmlcomment: Breaks ligatures in HTML comments.	244
v0.26		\LWR@includegraphicsb: Adapts to graphics syntax.	556
General: 2017/03/31	1	\LWR@newhtmlfile: If FormatEPUB or FormatWP: skips headers, footers, nav.	268
lwarp.css: Improved responsive marginpar and marginblock.	177		
cutwin: Added.	507		
endnotes: Added.	512		
floatflt: Added.	533		
footmisc: Added.	543		
footnotehyper: Added.	545		
footnote: Added.	544		
marginfix: Added.	590		
marginnote: Added.	591		

\LWR@parsetablecols: Added L, C, R, J column types.	330	\HTMLLanguage: Renamed from \MetaLanguage.	281
\LWR@startref: Removed space. . .	377	\HTMLPageBottom: Renamed from \SetPageBottom.	256
\chapter: If EPUB, prints footnotes before each section.	279	\HTMLPageTop: Renamed from \SetPageTop.	256
\hyperindexref: Improved indexing.	400	General: 2017/04/29	1
\textup: Fixed span class.	456	lwarmpm-newproject removed, and combined with lwarmpm.	175
General: 2017/04/14	1	lwarmpmk: Add: xdyfile configuration option.	217
glossaries: Added.	551	lwarmpmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lua ¹ latex.	217
graphics: Added.	552	lwarmpmk: Fix: xindy now used for print index generation with latexmk.	217
tabularx: Fix for optional pos. . .	669	lwarmpmk: language now used for both index and glossary generation.	217
tabulary: Added.	669	File: lwarmp_html.xdy renamed to lwarmp.xdy.	214
lwarmpmk: Add: printglossary and htmlglossary commands.	217	Fix: *.css files only written in print mode.	177
Added boolean FormatEPUB.	167	Fix: lwarmp.xdy only written in print mode.	214
Added boolean FormatWP.	167	Fix: lwarmp_mathjax.txt: Only written in print mode.	215
Added boolean HTMLDebugComments.	166	Option lwarmpmklang changed to IndexLanguage.	145
Added boolean HTMLMarkFloats, changed to WPMarkFloats as of v0.42.	168	Option OSWindows replaces macro \warpOSWindows.	146
Docs: Modfying lwarmpmk and index processing.	133	Option xdyFilename added.	145
File: lwarmp_mathjax.txt: Updated CDN repository.	215	Option latexmk replaces macro \UseLatexmk.	146
Forced oneside to maintain large right margin.	156	Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename.	146
v0.29		v0.31	
\LWR@includegraphicsb: Fix: Error when no optional arguments. . .	556	General: 2017/05/15	1
General: 2017/04/15	1	keyfloat: Improved compatibility. . .	574
lwarmpmk: Add: language option for config files.	217	v0.32	
Add: lwarmpmklang option for lwarmp.	145	\RequirePackage: Fix: Ignores blanks in package list.	161
Docs: Using a glossary	78	General: 2016/06/09	1
File: *.lwarmpmkconf: Add: language option for config files. . .	176	glossaries: Prevent error with \glo@name not defined.	401
File: lwarmpmk.conf: Add: language option for config files. . .	176		
v0.30			
\CSSFilename: Renamed from \NewCSS.	256		
\HTMLAuthor: Renamed from \HTMLauthor.	257		
\HTMLDescription: Renamed from \NewHTMLdescription.	257		
\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	255		

lwarpmk: Fix: <code>io.lines()</code> changed to <code>file:lines()</code> due to luatex changes.	217	\LWR@nullfonts: Improved font control.	458
v0.33		\LWR@restoreorigformatting: booktabs: Works inside <code>lateximage</code>	402
\HTMLAuthor: Fix: Provides empty default author if none given. . .	257	Improved font control.	402
\LWR@loadbefore: Fix: No \PackageError if already loaded.	149	\LWR@subhtmlclass: Moved optional argument in front of mandatory.	245
\LWR@parseatcolumn: Fix: Column alignment with leftmost @. . . .	326	\LWR@tabledatacolumn: booktabs: Works inside <code>lateximage</code>	363
\LWR@tabledatasinglecolumn: Fix: Macros in tabular could cause extra data cell.	339	\fboxBlock: Added.	452
\LWR@vspace: Add: <code>\vspace</code> nullified.	465	\makebox: Fix: Handles paren arg. .	450
\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &. . . .	322	General: 2017/08/08	1
General: 2017/07/10	1	babel-french: Adds fixed-width HTML spaces to punctuation. .	240
amsmath: Removed <code>fleqn</code> option. .	159	balance: Added.	492
fancyhdr: Fix: Optional args for \lhead, etc.	521	booktabs: Works inside <code>lateximage</code>	366, 495
Add: Tabular at and bang columns now have their own HTML columns.	316	boxedminipage2e: Added.	496
cleveref: Fix: Loaded \AtEndPreamble.	442	boxedminipage: Prevented. . . .	495
Fix: Incorrectly-inline math environments.	417	crop: Added.	506
New handling of & to localize catcode changes.	316	enumerate: Added.	513
v0.34		enumitem: Added, no longer required.	513
\@fnsymbol: Text symbols instead of math.	295	everyshi: Added.	516
\InlineClass: Moved optional argument in front of mandatory. .	247	fancybox: Added.	518
\LWR@htmldivclass: Moved optional argument in front of mandatory.	245	fancyvrb: Added, no longer required.	523
\LWR@htmlclass: Moved optional argument in front of mandatory.	245	figcaps: Added.	528
\LWR@htmlclassline: Moved optional argument in front of mandatory.	246	filecontents: Required. Patched for morewrites.	157
\LWR@htmlspanclass: Moved optional argument in front of mandatory.	243	floatpag: Added.	534
		flushend: Added.	540
		fullpage: Added.	549
		hyperxmp: Added.	571
		idxlayout: Added.	572
		marginfit: Added.	590
		mdframed: Improved <code>mdtheorem</code> patch.	600
		moreverb: Added.	604
		paralist: Added.	629
		pdfscape: Added.	629
		pdfsync: Added.	630
		prelim2e: Added.	631
		rotfloat: Added.	638
		savetrees: Added.	639
		shadow: Added.	649
		syntonly: Added.	668

titles: No longer required.	681	super/subscripts in a <code>lateximage</code>	402
titleref: Prevented.	685	<code>\LWR@section</code> : Improved spacing. . .	274
xmpinl: Added.	733	<code>\LWR@stoppars</code> : Extra HTML source space after paragraphs.	254
Added.	726	<code>\fbox</code> : Fix: Uses <code>\fboxrule</code> and <code>\fboxsep</code>	452
Docs: Horizontal space limitations.	1	<code>\framebox</code> : Fix: Handles width and horiz position.	451
Docs: Misplaced alignment character.	134	<code>\makebox</code> : Fix: Handles width and horiz position.	450
File: <code>lwarp_mathjax.txt</code> : Version change.	215	General: 2017/08/17	1
File: <code>README.txt</code> : updated.	1	<code>babel-french</code> : Adjustements for French variants, load order, footnotes, ellipses.	240
Fix: Added the <code>eqnarray</code> environments.	417	footnote: Extra HTML source space after paragraphs.	544
Improved font control.	455	<code>siunitx</code> : Fix for <code>babel-french</code>	435
Lists refactored to remove <code>enumitem</code> requirement.	306	<code>siunitx</code> : Improved symbol support.	653
Verbatim refactored to remove <code>fancyvrb</code> requirement.	301	transparent: Added.	706
<code>lateximage</code> : Fix: <code>lateximage</code> with <code>minipage</code> , <code>\parbox</code> , <code>\makebox</code> , <code>\fbox</code> , <code>\framebox</code> , <code>\raisebox</code> , <code>\scalebox</code> , <code>\reflectbox</code>	429	<code>upref</code> : Added.	711
<code>BlockClass</code> : Moved optional argument in front of mandatory.	247	<code>xcolor</code> : Added <code>\fcolorboxBlock</code> , <code>\colorboxBlock</code>	719
<code>fminipage</code> : Added.	453	<code>xcolor</code> : Fix: Background none in print mode.	719
<code>LWR@nestspan</code> : Fix: Minipages, <code>BlocksClass</code> , and lists inside a span.	242	<code>xcolor</code> : Refactored <code>\LWR@colorstyle</code>	723
<code>LWR@tabular</code> : <code>booktabs</code> : Works inside <code>lateximage</code>	367	<code>xcolor</code> : Uses <code>\fboxrule</code> and <code>\fboxsep</code>	719
v0.35		<code>xcolor</code> : <code>\fcolorbox</code> etc. now work inside <code>lateximage</code>	719
General: 2017/08/08	1	Docs: Reorganized: Special cases and limitations.	96
Fix: <code>\textbf</code> and related.	455	Source: Improved formatting. . . .	1
v0.36		<code>lateximage</code> : Footnotes appear in regular text instead of the <code>lateximage minipage</code>	429
<code>\LWR@HTMLsanitize</code> : Fix for <code>babel-french</code>	426	<code>LWR@tabular</code> : Fix for <code>babel-french</code> . .	367
<code>\LWR@HTMLsanitizeexpand</code> : Fix for <code>babel-french</code>	426	v0.37	
<code>\LWR@closeparagraph</code> : Extra HTML source space after paragraphs.	251	<code>\include</code> : Maintains independent aux files for HTML.	164
<code>\LWR@currenttextcolor</code> : Fix for <code>\rule</code> when <code>xcolor</code> not loaded.	466	General: 2017/08/19	1
<code>\LWR@footnotetext</code> : Extra HTML source space after paragraphs.	259	\TeX accents: Added.	173
Force HTML superscripts.	259	<code>babel-french</code> : Adjustment for load order.	240
<code>\LWR@nullfonts</code> : Fix: Filenames while using <code>MATHJAX</code>	458	color: Prevented.	504
<code>\LWR@restoreorigformatting</code> : <code>siunitx</code> : Improved		comment: Maintains independent cutfiles for print, HTML.	147
		<code>siunitx</code> : Improved symbol support.	653
		textcomp: Improved support. . . .	671

lwarpmk: Removes additional HTML aux files.	217	Supports authblk with <div>s of class oneauthor instead of tabular.	296, 692
File handles reorganized.	163	\AddSubtitlePublished: Added.	298
v0.38		\LWR@domulticolumn: Add: Optional vpos and # rows.	353
\@seccntformat: Added for appendix.	273	\LWR@restoreorigformatting: Appended with \appto instead of calling various macros.	402
\ForceHTMLPage: Added.	271	\LWR@tabledatacolumnntag: Don't start a data cell if see \TabularMacro.	363
\ForceHTMLTOC: Added.	271	\ResumeTabular: Added.	361
\LWR@section: \part* starts a new HTML page, for appendix.	274	\TabularMacro: Added.	361
Modified spacing, uses \numberline.	274	\multicolumnrow: Added.	360, 610
\numberline: Added trailing \quad.	395	\printauthor: Removed minipages.	293
\part: Fix with article class.	279	Supports authblk with <div>s of class oneauthor instead of tabular.	293
General: 2017/08/27	1	\thanksmarkseries: Removed minipage footnotes.	693
appendix: Added.	489	General: 2017/09/05	1
arabicfront: Added.	489	a4wide: Added.	475
caption2: Prevented.	501	a4: Added.	475
chappg: Added.	503	a5comb: Added.	475
color: Forces xcolor as well.	504	addlines: Added.	482
fix2col: Added.	529	anysize: Added.	488
fncychap: Added.	541	authblk: Added.	491
grffile: Added.	563	bigdelim: Added.	493
metalogo: Added.	602	bigstrut: Added.	494
nonumonpart: Added.	614	chngepage: Prevented.	502
nopageno: Added.	614	ebook: Added.	510
pagenote: Option page disabled.	628	fullwidth: Added.	549
realscripts: Added.	634	midpage: Added.	604
relsize: Added.	635	multirow: Add: New optional vpos argument.	608
romanbarpagenumber: Added.	637	multirow: Add: Supports left/right border for bigdelim.	608
romanbar: Added.	636	multirow: Fix: Long text argument.	608
scalefnt: Added.	639	supertabular: Added.	667
siunitx: Removed from lwarp core.	653	textarea: Added.	670
textcomp: Removed from lwarp core.	671	titling: Improved compatibility.	689
tocbibind: Added.	695	titling: Removed extraneous center environments.	690
xltxtra: Added.	732	typearea: Added.	708
lwarpmk: Added print1 and html1 actions.	217	xtabular: Added.	733
Added \markboth, \sloppy, etc.	238	zwpgelayout: Added.	735
Docs: Enhanced <i>Supported</i> <i>Features</i> table.	52	Docs: Reorganized tabular discussion.	113
Docs: Index, tocbibind.	106		
Docs: Starred sections.	104		
v0.39			
\@maketitle: titling version.	692		
Native L ^A T _E X version.	296		
Removed minipages.	296, 692		

Titlepage \published and \subtitle removed. \AddSubtitlePublished restores.	297	midfloat: Added.	604
titlepage: Clear pending footnotes.	291	multirow: Improved bigdelim borders.	608
Removed minipages.	291	pfnote: Added.	630
titlingpage: Clear pending footnotes.	690	quotchap: Added.	632
v0.40		sectsty: Added.	647
\@chapcntformat: Added for tocbibind, anonchap.	273	stabular: Added.	660
\LWR@HTMLhline: Added.	366	tabs: Added.	669
\LWR@includegraphicsb: Add: Full \graphicspath support.	556	textcomp: Additional symbols, improved XeLaTeX and LuaLaTeX support.	671
\LWR@nullfonts: Fix: Long arguments for expandable command.	458	tocbibind: Improved for \simplechapter.	695
\LWR@restoreorigformatting: Improved \TeX logos inside a lateximage.	402	xltxtra: Fix for \showhyphens with XeLaTeX.	732
Improved symbols inside a lateximage.	402	No longer preloads xfrac.	159
Nullified \InlineClass, etc. inside a lateximage.	402	v0.41	
\LWR@tabledatacolumnntag: Fix for bigdelim: \ldelim, \rdelim. ..	363	\LWR@addcmidruletrim: Add: \cmidrule trims.	344
\chapter: Added support for quotchap.	279	\LWR@clearmidrules: Add: \cmidrule trims.	342
\multicolumnrow: Fix: Adapts to older multirow and xparse. ...	360	\LWR@closetabledatacell: Add: Mute > for \bottomrule.	320
\simplechapterdelim: Added for tocbibind, anonchap.	273	Fix: At/bang column with \multirow.	320
\underline: Added.	461	Fix: Cancel < for \multicolumn. ...	320
General: 2017/09/25	1	\LWR@domulticolumn: Add: \cmidrule trims.	353
adjmulticol: Added.	481	Added vertical rules.	354
anonchap: Added.	488	\LWR@nullifyNoAutoSpacing: Fix: \NoAutoSpacing in a tabular with babel-french.	367
bigdelim: Improved documentation.	493	\LWR@parsebarcolumn: Added vertical rules.	328
cuted: Added.	507	\LWR@printatbang: Add: \cmidrule trims.	338
dblfnote: Added.	508	Add: Mute at and bang columns for \bottomrule.	338
fnpos: Added.	542	\LWR@printbartag: Added vertical rules.	338
graphics: Moved out of the lwarp core.	552	\LWR@subaddcmidruletrim: Added. ...	344
graphics: Restores \includegraphics and \DeclareGraphicsExtensions in a lateximage.	552	\LWR@subcmidrule: Add: \cmidrule trims.	343
graphicx: Moved out of the lwarp core.	563	\LWR@tabledatasinglecolumnntag: Add: \cmidrule trims.	339
grffile: Directly supported.	563	Add: Mute < for \bottomrule. ...	339
		\LWR@tabularfinishrow: Unfinished tabular rows automatically filled.	323

\mcolrowcell: Added for	
\multicolumrow cells.	365
General: 2017/10/07	1
multirow: Add: \cmidrule trims.	608
Added vertical rules.	609
Fix: < spec.	609
LWR@tabular: Fix: \NoAutoSpacing	
in a tabular with babel-french.	368
Improved rules.	369
v0.42	
\@ensuredmath: Improved	
\ensuremath.	411
\@textsubscript: Added.	460
\@textsuperscript: Added.	460
\LWR@HTMLhline: If FormatWP force	
explicit border.	366
\LWR@HTMLtextstyle: Added.	455
\LWR@addformatwpalignment: If	
FormatWP add explicit style for	
cell alignment.	346
\LWR@addrulewidth: If FormatWP	
force explicit border.	345
\LWR@amsmathbody: Fix: Numbering	
and naming AMS math	
environments.	429
\LWR@amsmathbodynumbered: Fix:	
Numbering and naming AMS	
math environments.	429
\LWR@domulticolumn: If FormatWP	
add cell alignment.	354
\LWR@doubledollar: If FormatWP	
print LaTeX expression.	408
Improved \ensuremath.	408
Improved line spacing with	
mathjax.	408
\LWR@floatbegin: If FormatWP add	
a text frame.	384
\LWR@floatend: If FormatWP add a	
text frame.	385
\LWR@hspace: If FormatWP add	
\quads.	465
\LWR@htmlmathlabel: If FormatWP	
print LaTeX expression.	416
\LWR@includegraphicsb: Fix:	
Filename expansion.	556
If FormatWP, use explicit size.	557
\LWR@remembertag: Fix: Numbering	
and naming AMS math	
environments.	428
\LWR@restoreorigformatting:	
Improved \ensuremath.	404
\LWR@rule: If FormatWP add	
\quads.	468
\LWR@singledollar: If FormatWP	
print LaTeX expression.	409
\LWR@subaddcmidruletrim: Opt if	
no rule given.	344
\LWR@tabledatasinglecolumn: If	
FormatWP add cell alignment.	340
\LaTeX: If FormatWP use explicit	
style.	470
\TeX: If FormatWP use explicit style.	470
\listoffigures: Added boolean	
WPMarkLOFT.	393
\listoftables: Added boolean	
WPMarkLOFT.	393
\marginpar: If FormatWP emulate a	
wrapfig.	263
\marginparBlock: If FormatWP	
emulate a wrapfig.	263
\tableofcontents: Added boolean	
WPMarkTOC.	392
\underline: If FormatWP, use	
explicit styles for \underline,	
etc.	461
General: 2017/10/30	1
\textbf and related: If FormatWP,	
use explicit styles for \textsc,	
etc.	455
algorithmicx: If FormatWP add	
\quads.	483
epigraph: If FormatWP add HTML	
styles.	514
fancybox: If FormatWP add HTML	
styles.	518
floatflt: Added width.	533
includegraphics: Fix: Class key.	556
keyfloat: If FormatWP add explicit	
HTML style.	575
moreverb: Simplified formatting of	
listings.	604
morewrites: Added.	606
multirow: If FormatWP add cell	
alignment.	609
overpic: Added.	627
realscripts: Fix for subscripts in a	
lateximage.	634
sidenotes: If FormatWP add	
explicit HTML style.	652

siunitx: Improved <code>\ensuremath</code>	653	<code>\LWR@maybeprintpendingfootnotes</code> :	
soul: If FormatWP, add explicit		Added FootnoteDepth.	262
styles.	657	<code>\LWR@nullfonts</code> : Fix: Nullify dollar	
textcomp: Improved		inside filenames.	458
<code>\interrobangdown</code>	671	<code>\LWR@parsetablecols</code> : Ignore	
wrapfig: If FormatWP add explicit		spaces in col spec.	332
HTML style.	718	<code>\LWR@printmccoltype</code> : Added	
Added boolean WPMarkLOFT.	168	vertical rules.	350
Added boolean WPMarkMath.	168	<code>\LWR@restoreorigformatting</code> : Fix:	
Added boolean		<code>\ref</code> in math or lateximage.	402
WPMarkMinipages.	168	Fix: Improved math inside	
Added boolean WPMarkTOC.	168	lateximages.	404
Added boolean WPTitleHeading.	169	<code>\LWR@section</code> : Fix: Expansion in	
Docs: Added support page.	2	comparison.	274
Docs: Improper <code>\prevdepth</code>	134	Fix: Math in section name.	276, 278
Docs: Reorganized math		Fix: Nullify fonts inside HTML	
limitations	107	comment.	275
File: <code>lwarp_mathjax.txt</code> :		<code>\TabularMacro</code> : <code>\newcommand</code>	
Updated siunitx script.	215	instead of <code>\relax</code> to fix	
Fix: Numbering and naming AMS		supertabular and xtab.	361
math environments.	427	<code>\href</code> : Made robust.	381
If FormatWP, shift section		<code>\nameref</code> : Made robust.	379
headings.	169	<code>\nolinkurl</code> : Made robust.	381
tabbing: Added.	304	<code>\url</code> : Made robust.	381
lateximage: Fix: Numbering and		General: 2017/11/08	1
naming AMS math		breakurl: Added.	496
environments.	430	hyperref: Made robust.	567, 568, 570
center: If FormatWP use explicit		hyperref: <code>\Gauge</code> added.	570
text-align.	433	luatodonotes: Added.	588
minipage: Added boolean		todonotes: Added.	704
WPMarkMinipages.	449	<code>LWR@currentautosec</code> : Added.	274
If FormatWP add a text frame.	448	Added FootnoteDepth.	259
eqnarray: Fix: Numbering and		Docs: HTML settings table.	85
naming AMS math		Docs: Reorganized HTML	
environments.	419	customization.	85
If FormatWP print LaTeX			
expression.	418	v0.44	
equation: If FormatWP print LaTeX		<code>\@currentlabelname</code> : Adjustment	
expression.	414	for koma-script.	372
<code>LWR@BlockClassWP</code> : Added to factor		<code>\HTMLTitle</code> : Added.	257
code.	248	<code>\LWR@addformatwpalignment</code> : Fix	
<code>LWR@figcaption</code> : If FormatWP forces		for multicolumn alignment if	
italic captions.	387	FormatWP.	346
<code>LWR@tabular</code> : If FormatWP force		<code>\LWR@backgroundcolor</code> : Added.	723
explicit border.	369	<code>\LWR@filestart</code> : Add <code>\HTMLTitle</code>	283
		Fix <code>\HTMLAuthor</code>	282
v0.43		<code>\LWR@listitem</code> : Added list and	
<code>\LWR@domulticolumn</code> : Fix for		trivlist.	308
vertical rules.	354	<code>\LWR@patchlists</code> : Added list and	
Fix: Multicolumn trim.	353, 354	trivlist.	311

\LWR@strresult: Fix:	
\providecommand.	318
\addcontentsline: Automatic	
\LWR@newfloatanchor.	389
\chapter: Add preamble for	
koma-script.	279
\marginparBlock: Added.	263
\nopagecolor: xcolor: Fix for	
\nopagecolor.	724
\part: Add preamble for	
koma-script.	279
\title: Added \thetitle.	257
General: 2017/11/22	1
AMS environments: Fix: Groups for	
lateximages.	420
algorithmicx: Improved comment	
symbol.	483
atbegshi: Added.	490
cancel: Added.	497
changepage: Additional options.	502
easy-todo: Added.	509
fancyref: Added.	522
fixmetodonotes: Added.	531
fixme: Added.	530
fontenc: Allowed after lwarp.	542
hang: Added.	563
ifoddpage: Added.	573
ltxtable: Added.	587
luatodonotes: Improved.	588
lwarp-patch-komascript: Added. .	737
overpic: Fix: Groups for	
lateximages.	627
pdfsync: Fixes.	630
preview: Added.	632
scrextend: Added.	639
scrhack: Added.	642
scrlayer-notecolumn: Added.	644
scrlayer-scrpage: Added.	645
scrlayer: Added.	643
section: Added.	646
soulpos: Added.	659
soulutf8: Added.	659
supertabular: Fix for caption.	667
tocbasic: Added.	694
tocloft: Added \newlistentry. .	701
tocloft: Improved \newlistof. .	702
tocstyle: Added.	703
todonotes: Improved.	704
todo: Added.	703
typearea: Added expert	
commands.	708
watermark: Added.	717
xcolor: Added	
\LWR@currenttextcolorstyle.	722
xcolor: Added	
\LWR@findcurrentcolor.	722
xcolor: Added	
\LWR@textcurrentcolor.	723
xtab: Fix for caption.	734
Adjustment for koma-script.	154, 157
If pdfLaTeX, require T1 and UTF8	
encoding.	141
picture: overpic: Fix: Groups for	
lateximages.	444
tikzpicture: tikz: Fix: Groups for	
lateximages.	681
list: Added list and trivlist. .	309
LWR@nestspan: Added list and	
trivlist.	242
v0.45	
\@currentHref: Added.	380
\@donoparitem: Modified for HTML.	306
\@item: Modified for HTML.	306
\@mklab: Modified for HTML.	306
\CSSFilename: Improved filenames	
with underscores.	256
\LWR@LwarpStart: Fix: Lateximages	
on incorrect pages with Mathjax.	286
\LWR@filenamenoblanks: Fix:	
Section names with \	266
Fix: Section names with	
underscores.	266
\LWR@includegraphicsb: Improved	
URLs with underscores.	556
\LWR@newautoidanchor: Fix: No	
anchor if frozen autoid.	385
\LWR@notmemoirloadafter: Added.	149
\LWR@printpendingmpfootnotes:	
Added.	262
\LWR@startref: Fix: Labels with	
underscores.	378
\LWR@subhyperref: Improved URLs	
with underscores.	380
\LWR@subhyperrefclass: Improved	
URLs with underscores.	380
\LWR@sublabel: Fix: Labels with	
underscores.	376
\LWR@tabledatacolumnntag: Fix:	
Empty line between rows.	365

<code>\chapter</code> : Add optional heading title for memoir.	279	Docs: Improved MiKTeX install instructions.	61
<code>\newpage</code> : Added.	462	Docs: Moved table so doesn't interfere with install docs.	60
<code>\nolinkurl</code> : Improved URLs with underscores.	381	File: <code>lwarp_mathjax.txt</code> : Allow MATHJAX inside tabbing.	215
<code>\normalmarginpar</code> : Added.	264	File: <code>lwarp_mathjax.txt</code> : Allow MATHJAX inside verse.	215
<code>\reversemarginpar</code> : Added.	264	Fix: Empty <code>sidetoc</code>	392
<code>\section</code> : Add optional heading title for memoir.	280	Improved filenames with underscores.	567
<code>\tableofcontents</code> : Fix: Empty <code>sidetoc</code>	392	Improved URLs with underscores.	496
Fix: Patch <code>\AtBeginDocument</code>	392	Improved: Robust <code>\</code> , <code>\</code> , and <code>\textellipsis</code> commands.	461
<code>\url</code> : Improved URLs with underscores.	381	Separate <code>LWR@thisautoidWP</code> for word processor <code><div>s</code>	385
General: 2018/01/14	1	<code>thebibliography</code> : Patched to emphasize titles.	401
array: Added.	490	<code>minipage</code> : Fix: Improper <code>\prevdepth</code>	447, 449
babel-french: Robust commands.	240		
backref: Added.	492	v0.46	
changebar: Added.	501	<code>\LWR@closeparagraph</code> : Fix: Tabular empty lines.	252
cite: Added.	503	<code>\LWR@closeprevious</code> : Fix: Stack unnesting.	238
continue: Added.	506	<code>\LWR@forcenewpage</code> : Fix: Improper <code>\prevdepth</code>	238
endfloat: Added.	511	<code>\LWR@lookforpackagename</code> : Fix: Spaces in <code>\usepackage</code>	161
epigraph: Support for memoir.	515	<code>\LWR@providelength</code> : Added.	142
fancyvrb: Improvements.	523, 524	<code>\LWRPrintStack</code> : Name changed from <code>\PrintStack</code>	237
flafter: Added.	532	<code>\makebox</code> : Fix: Lateximage in a <code>\makebox</code>	451
fltrace: Added.	540	<code>\popclose</code> : Fix: Stack unnesting.	231
footnpag: Added.	546	<code>\pushclose</code> : Fix: Stack unnesting.	230
fwlw: Added.	549	General: 2018/01/23	1
hanging: Added.	565	LWR@tabularpardepth added.	318
hyperref: Fix for underscores in <code>\hyperref</code>	567	amsthm: Adapted to <code>trivlist</code> changes.	486
lwarp-patch-memoir: Added.	739	mdframed: Fixes for svg math or lateximage in title.	596
memhfixc: Added.	602	mdframed: Fixes for footnotes.	597
memoir: Added.	473	ntheorem: Adapted to <code>trivlist</code> changes.	615
natbib: Added.	612	theorem: Adapt to <code>trivlist</code> changes.	677, 679
pagesel: Added.	628	list: Fix: Stack unnesting.	309
prettyref: Added.	631	LWR@tabular: Fix: Tabular empty lines.	371
subfigure: Added.	666		
subfig: Fix for subcaption end tag.	664		
subfig: Fix: Math in subcaptions.	662		
textfit: Added.	674		
titleref: Added.	685		
turnthepage: Added.	708		
Allows memoir's preloaded packages.	150		
Docs: xparse warnings.	122		
Docs: Fix for double hyphens.	65		
Docs: Improved install instructions.	65		

v0.47

\LWR@LwarpStart: Fix for SVG math in \nameref.	287
\LWR@WPcell: Fix: Line wrap at HTML hyphen.	346
\LWR@caption@begin: Fix: Argument passed to \LWR@origcaption@begin. . .	387
\LWR@createautosec: Fix: Line wrap at HTML hyphen.	272
\LWR@domulticolumn: Fix: Line wrap at HTML hyphen.	353
\LWR@floatbegin: Fix: Line wrap at HTML hyphen.	384
\LWR@htmlclosecomment: Add \mbox to prevent line breaks. .	244
\LWR@newautoidanchor: Fix: Line wrap at HTML hyphen.	385
\LWR@printopenlist: Fix: Line wrap at HTML hyphen.	306
\LWR@singledollar: Added SVG math image baseline adjust and em sizing.	409
Fix: Line wrap at HTML hyphen. .	410
\LWR@startref: Fix: Line wrap at HTML hyphen.	377
\LWR@sublabel: Fix: Line wrap at HTML hyphen.	376
\captionlistentry: Fix: Line wrap at HTML hyphen.	388
\hypertoc: Fix: Line wrap at HTML hyphen.	396
\hypertocfloat: Fix: Line wrap at HTML hyphen.	397
\makebox: Fix: Line wrap at HTML hyphen.	451
General: 2018/01/30	1
adjmulticol: Fix: Line wrap at HTML hyphen.	481
blowup: Added.	494
caption: Added.	498
caption: Also loads lwarp-caption. .	163
changepage: Fix for pagecheck macros.	502
endheads: Added.	511
epigraph: Fix: Line wrap at HTML hyphen.	514
hanging: Fix: Line wrap at HTML hyphen.	565

hang: Fix: Line wrap at HTML hyphen.	563
keyfloat: Fix for SVG math in captions.	574
midpage: Fix: Line wrap at HTML hyphen.	604
multirow: Fix: Line wrap at HTML hyphen.	608
multitoc: Added.	611
ntheorem: Fix: Line wrap at HTML hyphen.	619
realscripts: Fix: Line wrap at HTML hyphen.	634
scrextend: Fix: Line wrap at HTML hyphen.	639
sectionbreak: Added.	647
sidenotes: Fix for SVG math in captions.	651
subfig: Fix for SVG math in captions.	662
subfig: Fix: Support \nameref. . .	661
xurl: Added.	735
lwarpmk: pdfcrop: Removed hires option for improved crop accuracy.	217
lateximage: Added CSS style option. .	429
Fix: Line wrap at HTML hyphen. .	432
center: Fix: Line wrap at HTML hyphen.	433
minipage: Fix: Line wrap at HTML hyphen.	448
flushleft: Fix: Line wrap at HTML hyphen.	434
flushright: Fix: Line wrap at HTML hyphen.	434
enumerate: Fix: Line wrap at HTML hyphen.	310
itemize: Fix: Line wrap at HTML hyphen.	310
LWR@BlockClassWP: Fix: Line wrap at HTML hyphen.	248

v0.48

\@@@setcpageref: Fix for new v0.21 of cleveref.	443
\@@@setcref: Fix for new v0.21 of cleveref.	442
\@@@setcrefrange: Fix for new v0.21 of cleveref.	442
\@biblabel: Improved bibliography label.	401

<code>\@item: Honors \makelabel.</code>	306	<code>\rightline: Added.</code>	434
<code>\@maketitle: Fix: Errors with</code>		<code>\thempfootnote: Removed</code>	
<code> IEEEtran class.</code>	296	<code> \itshape.</code>	262
<code>\LWR@LwarpStart: Adjusted space</code>		General: 2018/02/14	1
<code> around captions.</code>	285	acronym: Added.	480
<code>\LWR@ProvidesPackageDrop: Fix:</code>		acro: Added.	478
<code> Options with braces.</code>	163	chapterbib: Added.	503
<code>\LWR@addtabularhrulecolor:</code>		colortbl: Added.	335, 346, 504
<code> colortbl: Added.</code>	347	fancyheadings: Prevented.	521
<code>\LWR@addtabularrulecolors:</code>		fancyref: Now directly supported.	522
<code> colortbl: Added.</code>	347	hypcap: Added.	566
<code>\LWR@closetabledatcell:</code>		hypernat: Added.	566
<code> colortbl: Added.</code>	321	hyperref: <code>\texorpdfstring</code> now	
<code>\LWR@includegraphicsb: Fix:</code>		uses the \TeX string.	569
<code> Virtual page size limited to a</code>		luatodonotes: Improved	
<code> group.</code>	556, 559	<code>\todotoc.</code>	588
<code>\LWR@lookforpackagename: Fix:</code>		siunitx: Changes fraction to	
<code> Parsing similar package names.</code>	160	symbol.	655
<code>\LWR@newautopagelabel: Fix: TOC,</code>		siunitx: Improved svg math.	653, 654
<code> LOF, LOT links.</code>	267	siunitx: Improved color output.	653
<code>\LWR@newhtmlfile: Fix: TOC, LOF,</code>		stfloats: Added.	660
<code> LOT links.</code>	269	todonotes: Improved	
<code>\LWR@nullfonts: Fix: \newline in</code>		<code>\todotoc.</code>	705
<code> title.</code>	458	vmargin: Added.	714
<code>\LWR@parsedrequirepackagenames:</code>		<code>xfrac: Fix: Added groups around</code>	
<code> Fix: Parsing similar package</code>		super/subscripts to localize	
<code> names.</code>	160	<code>LWR@nestspan</code> changes.	730
<code>\LWR@parsetablecols: Fix: Ignore</code>		Docs: Converting an existing	
<code> optional tabular column</code>		document.	80
<code> arguments.</code>	333	Improved font control.	458, 459
<code>\LWR@restoreorigformatting: Fix:</code>		<code>lateximage: Print mode boxed to</code>	
<code> \mbox in svg math, lateximage,</code>		natural width.	433
<code> Tikz.</code>	404	<code>abstract: Allow optional name.</code>	299
<code> Fix: Font sizes in svg math,</code>		<code>LWR@tabular: colortbl: Added.</code>	368
<code> lateximage, Tikz.</code>	402	v0.49	
<code> Fix: Spacing in svg math,</code>		<code>\LWR@addtabularcellcolor: xcolor:</code>	
<code> lateximage, Tikz.</code>	403	Added tabular row colors.	348
<code>\LWR@section: Fix: TOC, LOF, LOT</code>		<code>\LWR@domulticolumn: xcolor: Added</code>	
<code> links.</code>	278	tabular row colors.	354
<code>\LWR@tabledatasinglecolumn: tag:</code>		<code>\LWR@printlength: Fix: Group</code>	
<code> colortbl: Added.</code>	340	printlen changes.	159
<code>\LWR@textcurrentfont: Added.</code>		<code>\LWR@singledollar: Fix: Incorrect</code>	
<code> Improves font control.</code>	459	PDF output size corrupted HTML.	409
<code>\bibliography: Fix: \BaseJobname</code>		<code>\affiliation: Fix: Adapts to classes</code>	
<code> for bibliography.</code>	401	which already provide.	291
<code>\centerline: Added.</code>	434	<code>\href: Fix: Adapt to classes.</code>	381
<code>\l@part: Adapts to classes without</code>		<code>\noalign: Fix: \noalign inside</code>	
<code> \part.</code>	397	tabular.	366
<code>\leftline: Added.</code>	434	<code>\url: Fix: Adapt to classes.</code>	381
<code>\mbox: Nullified for HTML.</code>	450	General: 2018/02/19	1

amsmath: Fix: Patches for <code>\eqref</code> .	159	pagegrid: Added.	628
eso-pic: Fix for		pdfrender: Added.	630
<code>\AddToShipoutPicture</code> .	516	resizegather: Added.	636
figsize: Added.	529	vertbars: Added.	713
fnlineno: Added.	542	vwcol: Added.	714
hypdestopt: Added.	566	xcolor: Added tabular row	
hyphenat: Added.	571	colors.	335, 728
lineno: Added.	578	Fix: Adapt to classes.	462
luacolor: Added.	588		

§ 328 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>	407	<code>\@mpfootnotetext</code>	4204
<code>\&</code>	233, 5524	<code>\@nbitem</code>	5290
<code>\(</code>	7473	<code>\@opargbegintheorem</code>	5190
<code>\)</code>	7473	<code>\@rowc@lors</code>	5770
<code>\,</code>	96	<code>\@rowcolors</code>	5769
<code>\@@@setcpageref</code>	8234	<code>\@secCNTformat</code>	4499
<code>\@@@setcref</code>	8197	<code>\@starttoc</code>	7037
<code>\@@@setcrefrange</code>	8211	<code>\@textsubscript</code>	8650
<code>\@author</code>	289	<code>\@textsuperscript</code>	8646
<code>\@begintheorem</code>	5184	<code>\@title</code>	289
<code>\@biblabel</code>	7241	<code>\@wrglossary</code>	7218
<code>\@capttype</code>	6909	<code>\@wrindex</code>	7209
<code>\@chapcntformat</code>	4501	<code>\@</code>	462
<code>\@currentHref</code>	6805	<code>\$</code>	408
<code>\@currentlabelname</code>	6666	<code>\$\$</code>	408
<code>\@date</code>	289	<code>~</code>	96
<code>\@dlbfloat</code>	6881	A	
<code>\@donoparitem</code>	5210	<code>a4</code> (package)	475
<code>\@endtheorem</code>	5196	<code>a4wide</code> (package)	475
<code>\@ensuredmath</code>	7477	<code>a5comb</code> (package)	475
<code>\@float</code>	6881	<code>abstract</code> (environment)	5079
<code>\@fnsymbol</code>	4990	<code>abstract</code> (package)	104, 475
<code>\@footnotetext</code>	4202	<code>\abstractname</code>	87, 5078
<code>\@include</code>	579	accents	270
<code>\@item</code>	5223	<code>acro</code> (package)	478
<code>\@makecaption</code>	6911	<code>acronym</code> (package)	479
<code>\@makefnmark</code>	4173	adapting	
<code>\@makefntext</code>	4172	class	132
<code>\@maketitle</code>	49, 5013	document	80
<code>\@mklab</code>	5204	package	131
		<code>\addcontentsline</code>	6985

addlines (package)	482	\bibliography	7234
\AddSubtitlePublished	5041	\BibTeX	8854
adjmulticol (package)	481	bigdelim (package)	117, 492
Adobe (program)	57	bigstrut (package)	494
\affiliation	4893	BlockClass (environment)	3915
afterpackage (package)	157	\BlockClassSingle	3923
afterpage (package)	482	blowup (package)	494
algorithmic		bookmark (package)	495
with newfloat, trivfloat	707	booktabs (package)	495
algorithmicx (package)	482	boolean:	
algorithmx (package)	118	CombineHigherDepths	87, 265
align (environment)	7744	FileSectionNames	87, 234
align* (environment)	7767	FormatEpub	123, 167
alignment tab character &, misplaced		FormatWP	125, 167
.....	114, 313, 321	HTMLDebugComments	166
alltt (package)	484	LWR@amsmultline	416
\AmS	8859	LWR@copiedsidetoc	392
AMSMath		LWR@doingapar	250
split miss-numbered	110, 615	LWR@doingcmidrule	317
amsmath (package)	159	LWR@doinghline	317
amsthm (package)	484	LWR@doingstartpars	249
\and	289	LWR@doingtbrule	317
anonchap (package)	488	LWR@emptyatbang	317
anysize (package)	488	LWR@exitingtabular	362
appendix (package)	105, 489	LWR@freezethisautoid	385
arabicfront (package)	489	LWR@infloatrow	555
array (package)	490	LWR@intabularmetadata	318
\arrayrulecolor	5779	LWR@isstartingequation	427
AsciiDoc (program)	57	LWR@minipagefullwidth	446
AsciiDoctor (program)	57	LWR@minipagethispar	446
Asciidoctor-LaTeX (program)	57	LWR@opttablecol	330
atbegshi (package)	490	LWR@skipatbang	317
\attrib	120, 300, 711	LWR@skippingmcolrowcell	317
\attribution	5089	LWR@skippingmrowcell	317
authblk (package)	105, 491	LWR@starredlongtable	355
author		LWR@startedrow	316
HTML meta tag	95, 257	LWR@tableparcell	317
\author	94, 289	LWR@tabularcelladded	317
autosec	272	LWR@tabularmutemods	362
		LWR@tracinglwarp	166
		LWR@validtablecol	330
		LWR@verbtags	302
		mathjax	144
		usingOSWindows	143
		warpingHTML	144
		warpingprint	144
		WPMarkFloats	127, 167
		WPMarkLOFT	128, 168
		WPMarkMath	128, 168
		WPMarkMinipages	127, 168
B			
babel (package)	121		
\backmatter	4480		
backref (package)	492		
balance (package)	492		
\BaseJobname	3582		
BaseJobname (option)	84, 145		
baseline			
tabular	330		
\bfseries	8635		

project-specific changes	91	warpprint	89, 92
\CSSFilename	88, 90, 4128	environments:	
cuted (package)	507	abstract	5079
cutwin (package)	507	align	7744
		align*	7767
D		BlockClass	3915
danger icon	140	BVerbatim	196
\date	94	center	8031
dblfnote (package)	508	description	5365
dcolumn (package)	508	displaymath	7491
debugging	134	enumerate	5346
HTML debug comments	166	eqnarray	7599
\DeclareGraphicsExtensions	2	equation	7538
Deja Vu	81	equation*	7562
description		fcolorminipage	143, 8135
HTML meta tag	94, 257	flalign	7790
description (environment)	5365	flalign*	7813
displaymath (environment)	7491	flushleft	8047
document		flushright	8039
convert existing	80	fminipage	8463, 8510
\dotfill	8686	gather	7698
\doublerulesepcolor	5781	gather*	7721
draftwatermark (package)	509	itemize	5340
		lateximage	7946, 8026
E		list	5310
easy-todo (package)	509	longtable	2
ebook (package)	510	LWR@BlockClassWP	3939
ellipsis (package)	510	LWR@figcaption	6925
\emph	8519	LWR@nestspan	3779
emptypage (package)	511	LWR@tabular	6533
\end@dlbfloat	6897	LWRprint@fminipage	8486
\end@float	6897	math	7490
\EndDefiningTabulars	5518	minipage	8316
endfloat (package)	511	multline	7649
endheads (package)	511	multline*	7673
endnotes (package)	105, 512	picture	8297
\enlargethispage	8755	quote	5104
\enskip	463, 8709	tabbing	5172
enumerate (environment)	5346	thebibliography	7242
enumerate (package)	513	theindex	7187
enumitem (package)	513	tikzpicture	7
environ (package)	158	titlepage	4910
environment:		titlingpage	14
fminipage	99	verbatim	5162
lateximage	424	verse	2
picture	444	warpall	108
titlepage	93	warpHTML	58, 109
titlingpage	94	warpprint	109
warpall	92	epigraph (package)	514
warpHTML	89, 92	EPS images	111, 436

epstopdf (package)	111, 113, 436	project.lwarpmkconf	176
EPUB		project_html.tex	175
conversion software	123	sample_project.css	91, 214
HTML conversion settings . . .	123, 167	tutorial.tex	67
eqnarray (environment)	7599	filecontents (package)	157
equation		FileDepth (counter)	87, 265
miss-numbered	110, 615	FileSectionNames (boolean)	87, 234
equation (environment)	7538	fix2col (package)	529
equation* (environment)	7562	fixme (package)	122, 530
error messages	134	fixmetodonotes (package)	531
eso-pic (package)	515	flafter (package)	532
etoolbox (package)	142	\flagverse	712
everyhook (package)	157	flalign (environment)	7790
everypage (package)	516	flalign* (environment)	7813
everyshi (package)	516	Flare (program)	57
expl3 (package)	157	float (package)	118, 532
export		floatflt (package)	533
to word processor	125	floatpag (package)	534
extramarks (package)	516	floatrow (package)	118, 535
F			
fancybox (package)	100, 517	fltrace (package)	540
fancyhdr (package)	521	\flushbottom	3696
fancyheadings (package)	521	flushend (package)	540
fancyref (package)	522	flushleft (environment)	8047
fancyvrb (package)	523	flushright (environment)	8039
FAQ	134	fminipage (environment)	99, 8463, 8510
fbox		fncychap (package)	541
around a minipage	99	fnlineno (package)	542
\fbox	99, 8446	fnpos (package)	542
\fboxBlock	99, 8455, 8509	font	
\fcolorbox	91, 8099	bitmapped	81
\fcolorboxBlock	110, 8134	Computer Modern	81
fcolorminipage (environment) . . .	143, 8135	Deja Vu	81
figcaps (package)	528	ligatures	81, 82
figsize (package)	529	packages	81
file		selection	81
inaccessible	86, 87	size — lateximage	108, 406, 425
underscore	86, 97	size — math, SVG	108, 406, 425
file:		size — xfrac	730
glyphtounicode	82	fontenc (package)	82, 542
lwarp.css	91, 177	fontspec (package)	81, 543
lwarp.xdy	95, 214	footmisc (package)	543
lwarp_formal.css	209	footnote (package)	544
lwarp_mathjax.txt	215	FootnoteDepth (counter)	87, 259
lwarp_sagebrush.css	204	footnotehyper (package)	545
lwarp_tutorial.txt	67	footnotes	258
lwarpmk.conf	175	footnpag (package)	546
lwarpmk.lua	133	for	
project.css	91	HTML & PRINT	140
		HTML output	140
		PRINT output	140

lips (package)	581	\LWR@addmathjax	7527
list (environment)	5310	\LWR@addmulticolvertrulecolor ..	6190
list item, empty	97, 305	\LWR@addrulewidth	5996
listings (package)	581	\LWR@addtabularcellcolor	6093
\listof	7091	\LWR@addtabularhrulecolor	6057
\listoffigures	7065	\LWR@addtabulararrowcolor	6042
\listoftables	7078	\LWR@addtabularrulecolors	6069
lmodern (package)	81	\LWR@afterendverbatim	5146
lofdepth (counter)	396	\LWR@amsmathbody	7933
longtable (environment)	2	\LWR@amsmathbodynumbered	7939
longtable (package)	116, 585	LWR@amsmultline (boolean)	416
lotdepth (counter)	396	\LWR@atbeginverbatim	5131
lscap (package)	587	\LWR@backgroundcolor	48
ltcaption (package)	587	\LWR@beginhideamsmath	7584
ltxtable (package)	587	\LWR@blackborderpadding	8442
luacolor (package)	588	LWR@BlockClassWP (environment) ..	3939
LuaLaTeX		\LWR@botnavigation	3642
detection	155	\LWR@caption@begin	6945
file & section names	270	\LWR@caption@end	6957
\LuaLaTeX	8845	LWR@cellcolordepth (counter)	348
LuaLaTeX (program) [requirement] ..	60	\LWR@cellHTMLcolor	5774
\LuaTeX	8845	\LWR@clearmidrules	5928
luatodonotes (package)	121, 588	\LWR@closeparagraph	4020
lwarp		\LWR@closeprevious	3669
loading	83	\LWR@closetabledatcell	5474
options	83	\LWR@cmidrulewidth (length)	342
lwarp (package)	83	LWR@colafterspec	319
lwarp-patch-komascript (package) ..	736	LWR@colatspec	319
lwarp-patch-memoir (package)	738	LWR@colbangspec	319
lwarp.css (file)	91, 177	LWR@colbarspec	319
lwarp.xdy		LWR@colbeforespec	319
customizing	95	\LWR@columnHTMLcolor	5772
lwarp.xdy (file)	95, 214	\LWR@convertto	29
lwarp_formal.css (file)	209	LWR@copiedsidetoc (boolean)	392
lwarp_mathjax.txt (file)	215	\LWR@copyfile	614
lwarp_sagebrush.css (file)	204	\LWR@createautosec	4486
lwarp_tutorial.txt (file)	67	LWR@currentautosec (counter)	274
lwarpmk		\LWR@currentcss	4127
customizing	133	\LWR@currenttextcolor	8758
[lwarpmk]:		\LWR@descitem	5352
htmlglossary (option)	106, 551	\LWR@docmidrule	5959
printglossary (option)	106, 551	LWR@doingapar (boolean)	250
lwarpmk (option)	84, 145	LWR@doingcmidrule (boolean)	317
lwarpmk (program)	133, 217	LWR@doinghline (boolean)	317
lwarpmk.conf (file)	175	LWR@doingstartpars (boolean)	249
lwarpmk.lua (file)	133	LWR@doingtbrule (boolean)	317
\LWR@addcmidruletrim	5991	\LWR@domulticolumn	6204
\LWR@addcmidrulewidth	6022	\LWR@doubledollar	7390
\LWR@addformatwppalignment	6029	LWR@emptyatbang (boolean)	317
\LWR@addleftmostbartag	5853	\LWR@endhideamsmath	7592

<code>\LWR@endofline</code>	8692	<code>\LWR@htmlspanclass</code>	3822
<code>\LWR@ensuredoingapar</code>	3998	<code>\LWR@htmltag</code>	3832
<code>\LWR@equationtag</code>	7905	<code>\LWR@htmltagc</code>	3767
<code>LWR@exitingtabular (boolean)</code>	362	<code>\LWR@HTMLtextstyle</code>	8514
<code>LWR@externalfilecnt (counter)</code>	408	<code>\LWR@includegraphicsb</code>	100
<code>LWR@figcaption (environment)</code>	6925	<code>\LWR@indexitem</code>	7197, 7201, 7205
<code>\LWR@filenamenoblanks</code>	4306	<code>LWR@infloatrow (boolean)</code>	555
<code>\LWR@filestart</code>	4739	<code>\LWR@instertatbangcols</code>	5466
<code>\LWR@findword</code>	490	<code>LWR@intabularmetadata (boolean)</code>	318
<code>\LWR@floatbegin</code>	6860	<code>LWR@isstartingequation (boolean)</code>	427
<code>\LWR@floatend</code>	6883	<code>\LWR@itemizeitem</code>	5333
<code>\LWR@footnotetext</code>	4174	<code>LWR@latestautopage (counter)</code>	387
<code>\LWR@forceminwidth</code>	8431	<code>LWR@lateximagedepth (counter)</code>	425
<code>\LWR@forcenewpage</code>	3686	<code>\LWR@lateximagedepthref</code>	6696
<code>LWR@freezethisautoid (boolean)</code>	385	<code>LWR@lateximagenumber (counter)</code>	425
<code>\LWR@futurenonSPACElet</code>	5420	<code>\LWR@lateximagenumberref</code>	6699
<code>\LWR@FVstyle</code>	37	<code>\LWR@lateximagesfile</code>	576
<code>\LWR@getexparray</code>	3551	<code>\LWR@lightrulewidth (length)</code>	342
<code>\LWR@getmynexttoken</code>	5427	<code>LWR@LIpage (counter)</code>	425
<code>\LWR@heavyrulewidth (length)</code>	342	<code>\LWR@listitem</code>	5295
<code>\LWR@hidelatexequation</code>	7513	<code>\LWR@loadafter</code>	119
<code>\LWR@hspace</code>	8717	<code>\LWR@loadbefore</code>	133
<code>\LWR@HTMLarrayrulecolor</code>	5780	<code>\LWR@loadnever</code>	142
<code>\LWR@htmlblockcomment</code>	3865	<code>\LWR@longtabledatacaptiontag</code>	6254
<code>\LWR@htmlblocktag</code>	3867	<code>\LWR@lookforpackagename</code>	493
<code>\LWR@HTMLcellcolor</code>	5778	<code>\LWR@LwarpEnd</code>	4864, 8881
<code>\LWR@HTMLcline</code>	6518	<code>\LWR@LwarpStart</code>	4792, 8881
<code>\LWR@htmlclosecomment</code>	3837	<code>\LWR@maketitlesetup</code>	33, 4982
<code>\LWR@HTMLcolumncolor</code>	5776	<code>\LWR@maybenewtablerow</code>	5783
<code>\LWR@htmlcomment</code>	3858	<code>\LWR@maybeprintpendingfootnotes</code>	4246
<code>\LWR@htmldivclass</code>	3892	<code>LWR@midrulecounter (counter)</code>	342
<code>\LWR@htmldivclassend</code>	3895	<code>LWR@midrules</code>	341
<code>\LWR@HTMLdoublerulesepcolor</code>	5782	<code>LWR@minipagedepth (counter)</code>	446
<code>\LWR@html element</code>	3905	<code>LWR@minipagefullwidth (boolean)</code>	446
<code>\LWR@html elementclass</code>	3879	<code>LWR@minipageheight (length)</code>	446
<code>\LWR@html elementclassend</code>	3884	<code>\LWR@minipagestartpars</code>	8697
<code>\LWR@html elementclassline</code>	3898	<code>\LWR@minipagestoppars</code>	8700
<code>\LWR@html elementend</code>	3908	<code>LWR@minipagethispar (boolean)</code>	446
<code>LWR@htmlfilenumber (counter)</code>	235	<code>LWR@minipagewidth (length)</code>	446
<code>\LWR@htmlfileref</code>	6693	<code>\LWR@multicolother</code>	6140
<code>\LWR@HTMLhline</code>	6501	<code>\LWR@multicolpartext</code>	6135
<code>\LWR@htmlmathlabel</code>	7569	<code>\LWR@multicolskip</code>	6146
<code>\LWR@htmlmulticolumn</code>	6243	<code>\LWR@multirowborder</code>	3
<code>\LWR@htmlopencomment</code>	3837	<code>\LWR@mynexttoken</code>	5419
<code>\LWR@htmlrefsectionfilename</code>	3627	<code>\LWR@myshorttoc</code>	7009
<code>\LWR@HTMLrowcolor</code>	5777	<code>\LWR@nameref</code>	6690
<code>\LWR@HTMLsanitize</code>	7850	<code>LWR@nestspan (environment)</code>	3779
<code>\LWR@HTMLsanitizeexpand</code>	7871	<code>\LWR@newautoidanchor</code>	6903
<code>\LWR@htmlsectionfilename</code>	3594	<code>\LWR@newautopagelabel</code>	4379
<code>\LWR@htmlspan</code>	3814	<code>\LWR@newhtmlfile</code>	4387

\LWR@newlabel	6730	\LWR@rotboxorigin	238
\LWR@newref	6781	\LWR@rotstyle	86
LWR@nextautoid (counter)	386	\LWR@rowHTMLcolor	5773
LWR@nextautopage (counter)	386	\LWR@rule	8759
LWR@nextequation (counter)	412	\LWR@ruleHTMLcolor	5775
\LWR@nohspace	8745	\LWR@sanitize	3555
\LWR@notmemoirloadafter	130	\LWR@sanitized	3554
\LWR@nullfonts	8584	\LWR@scalestyle	89
\LWR@nullifyNoAutoSpacing	6520	\LWR@section	4503
\LWR@nulllistfills	5302	\LWR@sectionnumber	4483
\LWR@openparagraph	4003	\LWR@setexparray	3546
LWR@opttablecol (boolean)	330	\LWR@setlatestname	6668
\LWR@origcolspec	5460	\LWR@setOSWindows	37
\LWR@originname	239	\LWR@sidetoc	7104
\LWR@originnames	246	\LWR@singledollar	7408
\LWR@parseaftercolumn	5647	LWR@skipatbang (boolean)	317
\LWR@parseatcolumn	5593	LWR@skippingmcolrowcell (boolean) .	317
\LWR@parsebangcolumn	5617	LWR@skippingmrowcell (boolean) ...	317
\LWR@parsebarcolumn	5656	LWR@spandepth (counter)	249
\LWR@parsebeforecolumn	5638	\LWR@splabel	6702
\LWR@parseDcolumn	5683	LWR@starredlongtable (boolean) ...	355
\LWR@parsedrequirepackagenames ..	489	LWR@startedrow (boolean)	316
\LWR@parsenormalcolumn	5667	LWR@startingequation (counter) ...	427
\LWR@parsepcolumn	5679	\LWR@startingequationtag	7904
\LWR@parsetablecols	5689	\LWR@startnewdepth	4493
\LWR@patchlists	5371	\LWR@startpars	4071
LWR@prevFileDepth (counter)	273	\LWR@startref	6743
LWR@previousautopagelabel (counter)	267	\LWR@stoppars	4089
\LWR@printatbang	5829	\LWR@stripperiod	6667
\LWR@printbartag	5819	\LWR@strresult	5458
\LWR@printcloselist	5202	\LWR@subaddcmidruletrim	5978
\LWR@printlength	471	\LWR@subaddtabularcellcolor	6087
\LWR@printmccoldata	6149	\LWR@subcmidrule	5944
\LWR@printmccoltype	6114	\LWR@subhtml elementclass	3872
\LWR@printopenlist	5203	\LWR@subhyperref	6810
\LWR@printpendingfootnotes	4236	\LWR@subhyperrefclass	6824
\LWR@printpendingmpfootnotes ...	4255	\LWR@subinlineimage	6853
\LWR@printhetitle	4934	\LWR@sublabel	6707
\LWR@providelength	26	\LWR@subnewref	6775
\LWR@ProvidesPackageDrop	557	\LWR@subtableofcontents	7027
\LWR@ProvidesPackagePass	546	\LWR@subtabularhtmlcolumns	6321
\LWR@pushoneclose	4489	\LWR@syncmathjax	7493
\LWR@quickfile	573	LWR@tablecolindex (counter)	319
\LWR@remembertag	7909	\LWR@tablecolspec	5457
\LWR@requesttoc	4857	LWR@tablecolspecindex (counter) ...	319
\LWR@requirepackagenames	488	LWR@tablecolspecwidth (counter) ...	318
\LWR@restoreorigaccents	882	\LWR@tabledatacolumnntag	6398
\LWR@restoreorigformatting	7262	\LWR@tabledatasinglecolumnntag ..	5867
\LWR@restoreoriglists	5399	LWR@tableparcell (boolean)	317
\LWR@restoreorigprintxcolor	8173	LWR@tabletotalcols (counter)	319

a4wide	475	crop	506
a5comb	475	cuted	507
abstract	104, 475	cutwin	507
acro	478	dblfnote	508
acronym	479	dcolumn	508
addlines	482	draftwatermark	509
adjmulticol	481	easy-todo	509
afterpackage	157	ebook	510
afterpage	482	ellipsis	510
algorithmicx	482	emptypage	511
algorithmx	118	endfloat	511
alltt	484	endheads	511
amsmath	159	endnotes	105, 512
amsthm	484	enumerate	513
anonchap	488	enumitem	513
anysize	488	environ	158
appendix	105, 489	epigraph	514
arabicfront	489	epstopdf	111, 113, 436
array	490	eso-pic	515
atbegshi	490	etoolbox	142
authblk	105, 491	everyhook	157
babel	121	everypage	516
backref	492	everyshi	516
balance	492	expl3	157
bigdelim	117, 492	extramarks	516
bigstrut	494	fancybox	100, 517
blowup	494	fancyhdr	521
bookmark	495	fancyheadings	521
booktabs	495	fancyref	522
boxedminipage	495	fancyvrb	523
boxedminipage2e	496	figcaps	528
breakurl	496	figsize	529
calc	158	filecontents	157
cancel	497	fix2col	529
capt-of	389	fixme	122, 530
caption	118, 163, 389, 498	fixmetodonotes	531
caption2	501	flafter	532
ccaption	501	float	118, 532
changebar	501	floatflt	533
change page	502	floatpag	534
chappg	503	floatrow	118, 535
chapterbib	503	fltrace	540
chn gpage	502	flushend	540
cite	503	fncychap	541
cleveref	103, 441	fnlineno	542
cmap	82	fnpos	542
color	112, 504	fontenc	82, 542
colortbl	504	fontspec	81, 543
comment	147	footmisc	543
continue	506	footnote	544

footnotehyper	545	mdframed	102, 592
footnpag	546	memhfixc	602
framed	546	metalogo	602
ftnright	548	microtype	82, 155, 603
fullpage	549	midfloat	604
fullwidth	549	midpage	604
fwlw	549	moreverb	604
geometry	156, 550	morewrites	606
getttitlestring	157	mparhack	606
glossaries	105, 550	multicol	607
graphics	111, 552	multirow	608
graphicx	111, 563	multitoc	611
grffile	112, 563	nameref	611
hang	563	natbib	612
hanging	565	needspace	612
hypcap	566	newclude	121, 613
hypdestopt	566	newfloat	158
hypernat	566	newtxmath	111
hyperref	103, 380, 567	newunicodechar	82, 613
hyperxmp	571	nextpage	613
hyphenat	571	nicefrac	111
idxlayout	572	nonumonpart	614
ifoddpag	573	nopageno	614
ifplatform	142	nowidow	614
indentfirst	573	ntheorem	110, 615
inputenc	82, 574	overpic	113, 627
keyfloat	119, 574	pagegrid	628
kvoptions	144	pagenote	105, 628
layout	576	pagesel	628
letltxmacro	142	paralist	629
letterspace	577	parskip	629
lettrine	577	pdflscape	629
lineno	578	pdfrender	630
lips	581	pdfsync	630
listings	581	pfnote	630
lmodern	81	placeins	631
longtable	116, 585	prelim2e	631
lscap	587	prettyref	631
ltxcaption	587	preview	632
ltxtable	587	printlen	159
luacolor	588	quotchap	632
luatodonotes	121, 588	ragged2e	633
lwarp	83	realscripts	634
lwarp-patch-komascript	736	refcount	158
lwarp-patch-memoir	738	relsize	97, 635
makeidx	106, 158	resizegather	636
marginfit	590	romanbar	636
marginfix	590	romanbarpagenumber	637
marginnote	591	rotating	637
mcaption	591	rotfloat	638

savetrees	639	turnthepage	708
scaleft	639	typearea	708
scrextend	639	ulem	709
scrhack	642	units	111
scrlayer	642	upref	711
scrlayer-notecolumn	644	url	103
scrlayer-scrpage	645	varioref	103
section	646	varwidth	160
sectionbreak	647	verse	120, 711, 712
sectsty	647	vertbars	713
setspace	648	vmargin	714
shadow	649	vwcol	714
showidx	649	wallpaper	717
showkeys	650	watermark	717
sidecap	650	wrapfig	718
sidenotes	651	xcolor	112, 437, 719
siunitx	110, 435, 652	xfrac	729
soul	657	xifthen	158
soulpos	659	xltxtra	732
soulutf8	659	xmpinl	733
stabular	660	xparse	122, 156
stfloats	660	xpatch	142
subcaption	118	xstring	158
subfig	118, 661	xtab	117, 733
subfigure	666	xurl	735
supertabular	117, 667	zref	159
syntonly	668	zwpagelayout	735
tabls	669	page	
tabularx	669	inaccessible	86, 87
tabulary	669	\pagebreak	8751
textarea	670	\pagecolor	69
textcomp	82, 97, 670	pagegrid (package)	628
textfit	674	pagenote (package)	105, 628
textpos	675	\pagenumbering	3699
theorem	675	\pageref	6788
threeparttable	680	\pagerefPageFor	6787
tikz	680	pagesel (package)	628
titleps	681	\pagestyle	3691
titleref	685	Pandoc (program)	57
titlesec	685	\par	
titletoc	687	hooks	253
titling	105, 689	\paragraph	4714
tocbasic	693	paralist (package)	629
tocbibind	106, 107, 694, 695	\parbox	8398
tocloft	105, 696	\parsemulticolumnalignment	6176
tocstyle	703	parskip (package)	629
todo	703	\part	4645
todonotes	121, 704	PDF images	111, 436
transparent	706	pdfcrop (program) [requirement]	60
trivfloat	118, 707	pdfLaTeX (program) [requirement]	60

pdfLaTeX (program)	60	title page	93
pdfseparate (program)	60, 65	\sffamily	8637
pdftocairo (program)	60, 65	\sfrac	729
pdftotext (program)	60, 65	shadow (package)	649
perl (program)	66	showidx (package)	649
XeLaTeX (program)	60	showkeys (package)	650
\RequirePackage	521	sidecap (package)	650
\resizebox	297	sidenotes (package)	651
resizgather (package)	636	SideTOCDepth (counter)	86, 394
\ResumeTabular	6385	\sidetocname	7101
\reversemarginpar	4291	\simplechapterdelim	4500
\rightline	8057	siunitx	
\rmfamily	8636	with TeXMaths	128
romanbar (package)	636	siunitx (package)	110, 435, 652
romanbarpagenumber (package)	637	\sloppy	3697
\rotatebox	258	soul (package)	657
rotating (package)	637	soulpos (package)	659
rotfloat (package)	638	soulutf8 (package)	659
\rowcolor	5777	\sp	8643
\rownum	5768	space	
\rule	8807	horizontal	96
		between minipages	463
S		split	
sample_project.css (file)	91, 214	miss-numbered	110, 615
savetrees (package)	639	stabular (package)	660
\sb	8644	stack depths	138
\scalebox	277	\StartDefiningTabulars	5514
scalefnt (package)	639	stfloats (package)	660
scrextend (package)	639	subcaption (package)	118
scrhack (package)	642	subequations	
scrlayer (package)	642	MathJax	109, 406
scrlayer-notecolumn (package)	644	subfig (package)	118, 661
scrlayer-scrpage (package)	645	subfigure (package)	666
\scshape	8641	\subparagraph	4722
section		\subsection	4697
depths	138	\subsubsection	4705
heading, word processor	128	subtitle	297
missing	86, 87	supertabular (package)	117, 667
\section	4684	SVG	
section (package)	646	images showing as HTML	73, 136
sectionbreak (package)	647	math summary	108, 405
sectsty (package)	647	mathsvg option	145
\SetHTMLFileNumber	3585	syntonly (package)	668
setspace (package)	648		
settings		T	
css project-specific	91	tabbing (environment)	5172
CSS selection	90	\tableofcontents	89, 7046
HTML conversion	85	tabls (package)	669
lwrap package options	83	tabular	
selecting output	92	baseline	330

HTML columnn conversion	331	tikz (package)	680
macros inside	115, 314	tikzpicture (environment)	7
misplaced \noalign	116, 315	Title	
misplaced alignment tab character &		HTML meta tag	257
.....	114, 313	title	
multicolumn with multirow	610	HTML meta tag	95, 257
row corruption	115, 314	\title	94, 4138
\TabularMacro	6383	titlepage	
tabularx (package)	669	subtitle and published	297
tabulary (package)	669	titlepage (environment)	93, 4910
tagged equations		titles (package)	681
Mathjax	109, 406	titleref (package)	685
\TeX	8815	titles (option) [tocloft]	105
TeX2page (program)	56	titlesec (package)	685
TeX4ht (program)	56	titletoc (package)	687
TeXMaths (program)	128	titling (package)	105, 689
textarea (package)	670	titlingpage (environment)	14, 94
\textbf	8528	tocbasic (package)	693
\textcolor	60	[tocbibind]:	
textcomp (package)	82, 97, 670	numindex (option)	107, 695
textfit (package)	674	tocbibind (package)	106, 107, 694, 695
\textgreater	3575	tocdepth (counter)	86
\textit	8553	[tocloft]:	
\textless	3575	titles (option)	105
\textmd	8522	tocloft (package)	105, 696
\textnormal	8570	tocstyle (package)	703
textpos (package)	675	todo (package)	703
\textrm	8531	todonotes (package)	121, 704
\textsc	8556	\tracinglwrap	136, 641
\textsf	8538	transparent (package)	706
\textsl	8563	trivfloat	
\textsubscript	8647	with newfloat, algorithmic	707
\textsuperscript	8645	trivfloat (package)	118, 707
\texttt	8543	troubleshooting	134
\textup	8546	HTML debug comments	166
\tfl@chapter@fix	707	\ttfamily	8638
\thanks	94	TtH (program)	56
\thanksmarkseries	84	turnthepage (package)	708
\theauthor	290	tutorial.tex (file)	67
thebibliography (environment)	7242	typearea (package)	708
\thedate	290		
theindex (environment)	7187		
\thempfootnote	4233		
theorem (package)	675		
\thetitle	290		
\thispagestyle	3692		
threeparttable (package)	680		
tikz			
catcodes	681		
dollar redefined	681		

U

