



The **lwarp** package

\LaTeX to HTML

v0.57 — 2018/06/06

© 2016–2018 Brian Dunn
bd@BDTechConcepts.com

Abstract

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

Documents may be produced by 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. Almost 300 \LaTeX packages are supported with these high-level source compatibility replacements, and many others work as-is.

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

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

For a list of supported features, see [table 1: Supported packages and features](#).

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

License:

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

1 Supporting 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	45
	List of Tables	45
2	Updates	46
3	Introduction	61
3.1	Supported packages and features	63
4	Alternatives	68
4.1	Internet class	68
4.2	TeX4ht	68
4.3	Translators	68
4.4	AsciiDoc and AsciiDoctor	69
4.4.1	AsciiDoctor-LaTeX	69
4.5	Pandoc	69
4.6	Word processors	69
4.7	Commercial systems	69
4.8	Comparisons	69
5	Installation	71
5.1	Installing the lwarp package	73
5.2	Installing the lwarpmk utility	75
5.2.1	Using a local copy of lwarpmk	76
5.3	Installing additional utilities	77
6	Tutorial	79
6.1	Starting a new project	79
6.2	Compiling the print version with lwarpmk	83
6.3	Compiling the HTML version with lwarpmk	84
6.4	Generating the SVG images	85
6.5	Using MATHJAX for math	86
6.6	Changing the CSS style	87
6.7	Customizing the HTML output	87
6.8	Using latexmk	88
6.9	Using XeLaTeX or LuaLaTeX	89
6.10	Using a glossary	90
6.11	Cleaning auxiliary files	91
6.12	Cleaning auxiliary and output files	91
6.13	Cleaning the images from the lateximages directory	91
6.14	Converting PDF images to SVG	91
6.15	Creating HTML from an incomplete compile	91
6.16	Processing multiple projects in the same directory	91

6.17	Using the make utility	92
7	Converting an existing document	93
8	Additional details	94
8.1	Shell escape	94
8.2	Font and UTF-8 support	94
8.2.1	Indexes, glossaries, and encoding	95
8.3	lwarp package loading and options	96
8.4	Customizing the HTML output	98
8.4.1	Example HTML file naming	103
8.5	Customizing the CSS	104
8.6	Selecting the operating system	105
8.7	Selecting actions for print or HTML output	105
8.8	Commands to be placed into the warpprint environment	107
8.9	Title page	107
8.10	HTML page meta descriptions	108
8.11	HTML page meta title	108
8.12	HTML page meta author	109
8.13	Modifying xindy index processing	109
9	Special cases and limitations	109
9.1	Things to avoid	110
9.2	Formatting	110
9.2.1	Text formatting	110
9.2.2	Horizontal space	110
9.2.3	Text alignment	111
9.2.4	Accents	111
9.2.5	textcomp package	111
9.2.6	Superscripts and other non-math uses of math mode	111
9.2.7	Empty <code>\item</code> followed by a new line of text or a nested list:	111
9.2.8	Filenames and URLs in lists or footnotes	112
9.2.9	relsize package	112
9.3	Boxes and minipages	112
9.3.1	Marginpars	112
9.3.2	Save Boxes	112
9.3.3	Minipages	113
9.3.4	Side-by-side minipages	113
9.3.5	Framed minipages and other environments	113
9.3.6	fancybox package	115
9.3.7	mdframed package	116
9.4	Cross-references	117
9.4.1	Page references	117
9.4.2	cleveref and varioref packages	117
9.4.3	Hyperlinks, hyperref , and url	117
9.4.4	Footnotes and page notes	118

9.5	Front and back matter	119
9.5.1	Custom classes with multiple authors and affiliations	119
9.5.2	Starred chapters and sections	119
9.5.3	abstract package	120
9.5.4	titling and authblk	120
9.5.5	tocloft package	120
9.5.6	appendix package	120
9.5.7	pagenote package	121
9.5.8	endnotes package	121
9.5.9	BibTeX	121
9.5.10	glossaries package	121
9.5.11	Index and the tocbibind package	122
9.6	Math	123
9.6.1	Rendering tradeoffs	123
9.6.2	SVG option	124
9.6.3	MATHJAX option	125
9.6.4	Customizing MATHJAX	125
9.6.5	MATHJAX limitations	125
9.6.6	Catcode changes	126
9.6.7	Dynamic math	127
9.6.8	Display math	127
9.6.9	chemformula package	127
9.6.10	mhchem package	127
9.6.11	ntheorem package	128
9.6.12	siunitx package	128
9.6.13	units and nicefrac packages	128
9.6.14	newtxmath package	128
9.7	Graphics	129
9.7.1	tikz package	130
9.7.2	grffile package	130
9.7.3	color package	130
9.7.4	xcolor package	130
9.7.5	epstopdf package	131
9.7.6	overpic package	131
9.8	Tabbing	131
9.9	Tabular	131
9.9.1	longtable package	134
9.9.2	supertabular and xtab packages	135
9.9.3	bigdelim package	135
9.10	Floats	136
9.10.1	Float contents alignment	136
9.10.2	float , trivfloat , and/or algorithmicx together	136
9.10.3	caption and subcaption packages	136
9.10.4	subfig package	136
9.10.5	floatrow package	137
9.10.6	keyfloat package	137

9.11	Koma-Script	137
9.12	Memoir	138
9.13	Miscellaneous	138
9.13.1	verse and memoir	138
9.13.2	newclude package	139
9.13.3	babel package	139
9.13.4	polyglossia package	140
9.13.5	todonotes and luatodonotes packages	140
9.13.6	fixme	140
9.13.7	xparse	141
10	Epub conversion	142
11	Word-processor conversion	144
11.1	Activating word-processor conversion	144
11.2	Additional modifications	146
11.3	Recommendations	147
11.4	Limitations	148
12	Modifying lwarp	150
12.1	Modifying a package for lwarp	150
12.1.1	Adding a package to the lwarp.dtx file	151
12.2	Modifying a class for lwarp	152
12.3	Testing lwarp	152
12.4	Modifying lwarpmk	152
13	Troubleshooting	154
13.1	Using the lwarp.sty package	154
13.1.1	Debug tracing output	158
13.2	Compiling the lwarp.dtx file	158
1	lwarp.sty	159
14	Implementation	159
15	Section depths and HTML headings	160
16	Source Code	161
17	Detecting the TeX Engine — pdfflatex, lualatex, xelatex	162
18	MD5 hashing	162
19	pdfLaTeX T1 and UTF8 encoding	163
20	Unicode input characters	163

21	Miscellaneous tools	164
22	Early package requirements	164
23	Operating-System portability	165
23.1	Common portability code	165
23.2	Unix, Linux, and Mac OS	165
23.3	MS-WINDOWS	165
24	Package options	166
24.1	Conditional compilation	169
25	Package load order	170
25.1	Tests of package load order	171
25.2	Error for disallowed packages loaded before lwarp	172
25.3	Enforcing package loading after lwarp	172
26	Required packages	178
27	Loading packages	183
28	Additional required packages	186
29	File handles	187
30	Include a file	187
31	Copying a file	188
32	Debugging messages	189
33	Defining print and HTML versions of macros and environments	190
34	HTML-conversion output modifications	195
34.1	User-level controls	195
34.2	Heading adjustments	197
35	Remembering original formatting macros	198
36	Accents	201
37	Configuration Files	202
37.1	project_html.tex	202
37.2	lwarpmk.conf	203
37.3	project.lwarpmkconf	204
37.4	lwarp.css	205
37.5	lwarp_sagebrush.css	234
37.6	lwarp_formal.css	239

37.7	sample_project.css	244
37.8	lwarp.xdy	244
37.9	lwarp_one_limage.cmd	245
37.10	lwarp_mathjax.txt	245
37.11	lwarpmk.lua — lwarpmk option	248
38	Stacks	266
38.1	Assigning depths	267
38.2	Closing actions	267
38.3	Closing depths	268
38.4	Pushing and popping the stack	268
39	Data arrays	270
40	Localizing catcodes	270
41	Localizing dynamic math	272
42	Sanitizing labels and filenames	273
43	HTML entities	273
44	HTML filename generation	274
45	Homepage link	277
46	\LWRPrintStack diagnostic tool	277
47	Closing stack levels	278
48	PDF pages and styles	279
49	HTML tags, spans, divs, elements	280
49.1	Mapping \LaTeX Sections to HTML Sections	280
49.2	Babel-French tag modifications	281
49.3	HTML tags	282
49.4	Block tags and comments	284
49.5	Div class and element class	285
49.6	Single-line elements	286
49.7	HTML5 semantic elements	287
49.8	High-level block and inline classes	287
49.9	Closing HTML tags	289
50	Paragraph handling	290
51	Paragraph start/stop handling	294
52	Page headers and footers	297

53	CSS	297
54	Title, HTML meta author, HTML meta description	298
55	Footnotes	299
55.1	Regular page footnotes	300
55.2	Minipage footnotes	300
55.3	Titlepage thanks	300
55.4	Regular page footnote implementation	300
55.5	Minipage footnote implementation	303
55.6	Printing pending footnotes	304
56	Marginpars	305
57	Splitting HTML files	306
58	Sectioning	314
58.1	User-level starred section commands	315
58.2	Book class commands	315
58.3	Sectioning support macros	316
58.4	\section and friends	323
59	Starting a new file	325
60	Starting HTML output	328
61	Ending HTML output	331
62	Title page	333
62.1	Setting the title, etc.	333
62.2	\if@titlepage	335
62.3	Changes for \affiliation	335
62.4	Printing the thanks	336
62.5	Printing the title, etc. in HTML	336
62.6	Printing the title, etc. in print form	338
62.7	\maketitle for HTML output	338
62.8	\published and \subtitle	342
63	Abstract	343
64	Quote and verse	344
64.1	Attributions	344
64.2	Quotes, quotations	344
64.3	Verse	345
64.3.1	L ^A T _E X core verse environment	345
64.3.2	verse and memoir	346
65	Verbatim and tabbing	346

66	Theorems	349
67	Lists	350
67.1	List environment	351
67.2	Itemize	355
67.3	Enumerate	355
67.4	Description	356
67.5	Patching the lists	357
68	Tabular	358
68.1	Limitations	358
68.2	Temporary package-related macros	361
68.2.1	arydshln	361
68.3	Token lookahead	361
68.4	Tabular variables	362
68.5	Handling &, @, !, and bar	365
68.5.1	Handling &	367
68.5.2	Filling an unfinished row	368
68.6	Handling \	370
68.7	Looking ahead in the column specifications	371
68.8	Parsing @, >, <, !, bar columns	372
68.9	Parsing 'l', 'c', or 'r' columns	376
68.10	Parsing 'p', 'm', or 'b' columns	376
68.11	Parsing 'w' columns	377
68.12	Parsing 'D' columns	377
68.13	Parsing the column specifications	377
68.14	colortbl and xparse tabular color support	382
68.15	Starting a new row	384
68.16	Printing vertical bar tags	385
68.17	Printing at or bang tags	385
68.18	Data opening tag	386
68.19	Midrules	388
68.20	Cell colors	395
68.21	Multicolumns	399
68.21.1	Parsing multicolumns	399
68.21.2	Multicolumn factored code	403
68.21.3	Multicolumn	406
68.21.4	Longtable captions	406
68.21.5	Counting HTML tabular columns	409
68.22	Multirow if not loaded	411
68.23	Multicolumnrow	411
68.24	Utility macros inside a table	412
68.25	Special-case tabular markers	413
68.26	Checking for a new table cell	414
68.27	\mrowcell	417
68.28	\mcolrowcell	417

68.29	HTML tabular environment	417
69	Cross-references	424
69.1	Setup	424
69.2	Zref setup	426
69.3	Labels	427
69.4	References	429
69.5	Hyper-references	431
70	Floats	435
70.1	Float environment	436
70.2	Float tracking	438
70.3	Caption inside a float environment	439
70.4	Caption and LOF linking and tracking	440
71	Table of Contents, LOF, LOT	443
71.1	Reading and printing the TOC	444
71.2	High-level TOC commands	447
71.3	Side TOC	447
71.4	Low-level TOC line formatting	449
72	Index and glossary	452
73	Bibliography presentation	454
74	Restoring original formatting	456
75	Math	458
75.1	Limitations	458
75.1.1	Rendering tradeoffs	458
75.1.2	SVG option	459
75.1.3	MATHJAX option	459
75.1.4	Customizing MATHJAX	460
75.1.5	MATHJAX limitations	460
75.1.6	Catcode changes	461
75.1.7	Dynamic math	461
75.1.8	Display math	461
75.2	HTML alt tag names	462
75.3	Inline and display math	462
75.4	MATHJAX support	474
75.5	Equation environment	476
75.6	<code>\displaymathnormal</code> and <code>\displaymathother</code>	479
75.7	AMS Math environments	480
75.7.1	Support macros	480
75.7.2	Environment patches	481
76	Lateximages	490

76.1	Description	490
76.2	Support counters and macros	491
76.3	Font size	492
76.4	Sanitizing math expressions for HTML	493
76.5	Equation numbers	494
76.6	HTML alt tags	496
76.7	lateximage environment	497
77	center, flushleft, flushright	502
78	Pre-loaded packages	504
79	Siunitx	505
80	Graphics print-mode modifications	506
80.1	General limitations	506
80.2	Print-mode modifications	507
81	Xcolor boxes	507
82	Chemmacros environments	511
83	Cleveref	512
84	Picture	515
85	Boxes and Minipages	516
85.1	Counters and lengths	516
85.2	Footnote handling	517
85.3	Minipage handling	517
85.4	Parbox, mbox, makebox, framebox, fbox, raisebox	521
86	Direct formatting	526
87	Skips, spaces, font sizes	533
88	\phantomsection	541
89	\LaTeX and other logos	541
89.1	HTML logos	542
89.2	Print logos	544
90	\AtBeginDocument, \AtEndDocument	544
91	Koma-script	545
92	Memoir	545

93	Trademarks	546
2	lwarp-a4.sty	547
94	a4	547
3	lwarp-a4wide.sty	547
95	a4wide	547
4	lwarp-a5comb.sty	547
96	a5comb	547
5	lwarp-abstract.sty	547
97	abstract	547
6	lwarp-acro.sty	550
98	acro	550
7	lwarp-acronym.sty	551
99	acronym	551
8	lwarp-adjmulticol.sty	553
100	adjmulticol	553
9	lwarp-addlines.sty	554
101	addlines	554
10	lwarp-ae.sty	554
102	ae	554

11	lwarp-aecc.sty	554
103	aecc	554
12	lwarp-afterpage.sty	555
104	afterpage	555
13	lwarp-algorithm2e.sty	555
105	algorithm2e	555
14	lwarp-algorithmicx.sty	559
106	algorithmicx	559
15	lwarp-alltt.sty	560
107	alltt	560
16	lwarp-amsthm.sty	561
108	amsthm	561
17	lwarp-anonchap.sty	565
109	anonchap	565
18	lwarp-anysize.sty	565
110	anysize	565
19	lwarp-appendix.sty	566
111	appendix	566
20	lwarp-arabicfront.sty	566
112	arabicfront	566

21	lwarp-array.sty	567
113	array	567
22	lwarp-arydshln.sty	567
114	arydshln	567
23	lwarp-atbegshi.sty	569
115	atbegshi	569
24	lwarp-authblk.sty	570
116	authblk	570
25	lwarp-axodraw2.sty	571
117	axodraw2	571
26	lwarp-backref.sty	571
118	backref	571
27	lwarp-balance.sty	571
119	balance	571
28	lwarp-bibunits.sty	572
120	bibunits	572
29	lwarp-bigdelim.sty	572
121	bigdelim	572
30	lwarp-bigstrut.sty	573
122	bigstrut	573

31	lwarp-blowup.sty	574
123	blowup	574
32	lwarp-bookmark.sty	574
124	bookmark	574
33	lwarp-booktabs.sty	575
125	booktabs	575
34	lwarp-boxedminipage.sty	576
126	boxedminipage	576
35	lwarp-boxedminipage2e.sty	577
127	boxedminipage2e	577
36	lwarp-breakurl.sty	577
128	breakurl	577
37	lwarp-bytefield.sty	578
129	bytefield	578
38	lwarp-cancel.sty	578
130	cancel	578
39	lwarp-caption.sty	579
131	caption	579
40	lwarp-caption2.sty	582
132	caption2	582

41	lwarp-ccaption.sty	582
133	ccaption	582
42	lwarp-changebar.sty	583
134	changebar	583
43	lwarp-changepage.sty	583
135	changepage	583
44	lwarp-chngpage.sty	584
136	chngpage	584
45	lwarp-chappg.sty	584
137	chappg	584
46	lwarp-chapterbib.sty	584
138	chapterbib	584
47	lwarp-chemfig.sty	585
139	chemfig	585
48	lwarp-chemformula.sty	586
140	chemformula	586
49	lwarp-chemgreek.sty	591
141	chemgreek	591
50	lwarp-chemmacros.sty	592

142 chemmacros	592
142.1 Changes to the user's document	592
142.2 Code	593
142.3 Loading modules	593
142.4 New environments	593
142.5 Acid-base	594
142.6 Charges	596
142.7 Nomenclature	596
142.8 Particles	598
142.9 Phases	599
142.10 Mechanisms	600
142.11 Newman	602
142.12 Orbital	603
142.13 Reactions	603
142.14 Redox	604
142.15 Scheme	605
142.16 Spectroscopy	605
142.17 Thermodynamics	610
51 lwarp-chemnum.sty	612
143 chemnum	612
52 lwarp-cite.sty	613
144 cite	613
53 lwarp-clrdblpg.sty	613
145 clrdblpg	613
54 lwarp-color.sty	613
146 color	613
55 lwarp-colortbl.sty	614
147 colortbl	614
56 lwarp-continue.sty	616

148	continue	616
57	lwarp-crop.sty	616
149	crop	616
58	lwarp-cuted.sty	616
150	cuted	616
59	lwarp-cutwin.sty	617
151	cutwin	617
60	lwarp-dblfloatfix.sty	618
152	dblfloatfix	618
61	lwarp-dblfnote.sty	618
153	dblfnote	618
62	lwarp-dcolumn.sty	618
154	dcolumn	618
63	lwarp-diagbox.sty	619
155	diagbox	619
64	lwarp-draftwatermark.sty	620
156	draftwatermark	620
65	lwarp-easy-todo.sty	621
157	easy-todo	621

66	lwarp-ebook.sty	622
158	ebook	622
67	lwarp-ellipsis.sty	622
159	ellipsis	622
68	lwarp-emptypage.sty	623
160	emptypage	623
69	lwarp-endfloat.sty	623
161	endfloat	623
70	lwarp-endheads.sty	623
162	endheads	623
71	lwarp-endnotes.sty	624
163	endnotes	624
72	lwarp-enumerate.sty	625
164	enumerate	625
73	lwarp-enumitem.sty	625
165	enumitem	625
74	lwarp-epigraph.sty	626
166	epigraph	626
75	lwarp-epstopdf.sty	627
167	epstopdf	627

76	lwarp-epstopdf-base.sty	628
168	epstopdf-base	628
77	lwarp-errata.sty	628
169	errata	628
78	lwarp-eso-pic.sty	630
170	eso-pic	630
79	lwarp-everypage.sty	631
171	everypage	631
80	lwarp-everyshi.sty	631
172	everyshi	631
81	lwarp-extramarks.sty	631
173	extramarks	631
82	lwarp-fancybox.sty	632
174	fancybox	632
83	lwarp-fancyheadings.sty	638
175	fancyheadings	638
84	lwarp-fancyhdr.sty	638
176	fancyhdr	638
85	lwarp-fancyref.sty	639

177	fancyref	639
86	lwarp-fancyvrb.sty	640
178	fancyvrb	640
87	lwarp-figcaps.sty	646
179	figcaps	646
88	lwarp-figsize.sty	647
180	figsize	647
89	lwarp-fix2col.sty	647
181	fix2col	647
90	lwarp-fixme.sty	647
182	fixme	647
91	lwarp-fixmetodonotes.sty	649
183	fixmetodonotes	649
92	lwarp-flafter.sty	649
184	flafter	649
93	lwarp-float.sty	650
185	float	650
94	lwarp-floatflt.sty	651
186	floatflt	651

95	lwarp-floatpag.sty	652
187	floatpag	652
96	lwarp-floatrow.sty	653
188	floatrow	653
97	lwarp-fltrace.sty	658
189	fltrace	658
98	lwarp-flushend.sty	659
190	flushend	659
99	lwarp-fncychap.sty	659
191	fncychap	659
100	lwarp-fnlineno.sty	660
192	fnlineno	660
101	lwarp-fnpos.sty	660
193	fnpos	660
102	lwarp-fontenc.sty	661
194	fontenc	661
103	lwarp-fontspec.sty	661
195	fontspec	661
104	lwarp-footmisc.sty	661
196	footmisc	661

105	lwarp-footnote.sty	662
197	footnote	662
106	lwarp-footnotehyper.sty	664
198	footnotehyper	664
107	lwarp-footnpag.sty	664
199	footnpag	664
108	lwarp-forest.sty	664
200	forest	664
109	lwarp-framed.sty	665
201	framed	665
110	lwarp-ftnright.sty	667
202	ftnright	667
111	lwarp-fullpage.sty	668
203	fullpage	668
112	lwarp-fullwidth.sty	668
204	fullwidth	668
113	lwarp-fwlw.sty	668
205	fwlw	668
114	lwarp-geometry.sty	669
206	geometry	669

115	lwarp-glossaries.sty	669
207	glossaries	669
116	lwarp-glossary.sty	671
208	glossary	671
117	lwarp-graphics.sty	672
209	graphics	672
209.1	Graphics extensions	672
209.2	Length conversions and graphics options	672
209.3	Printing HTML styles	675
209.4	\includegraphics	676
209.5	Boxes	680
118	lwarp-graphicx.sty	683
210	graphicx	683
119	lwarp-grffile.sty	684
211	grffile	684
120	lwarp-grid.sty	684
212	grid	684
121	lwarp-gridset.sty	684
213	gridset	684
122	lwarp-hang.sty	685
214	hang	685
123	lwarp-hanging.sty	686

215	hanging	686
124	lwarp-hypcap.sty	687
216	hypcap	687
125	lwarp-hypdestopt.sty	688
217	hypdestopt	688
126	lwarp-hypernat.sty	688
218	hypernat	688
127	lwarp-hyperref.sty	688
219	hyperref	688
128	lwarp-hyperxmp.sty	694
220	hyperxmp	694
129	lwarp-hyphenat.sty	694
221	hyphenat	694
130	lwarp-idxlayout.sty	695
222	idxlayout	695
131	lwarp-ifoddpage.sty	696
223	ifoddpage	696
132	lwarp-indentfirst.sty	696
224	indentfirst	696

133	lwarp-inputenc.sty	697
225	inputenc	697
134	lwarp-inputenx.sty	697
226	inputenx	697
135	lwarp-keyfloat.sty	697
227	keyfloat	697
136	lwarp-layout.sty	699
228	layout	699
137	lwarp-letterspace.sty	700
229	letterspace	700
138	lwarp-lettrine.sty	700
230	lettrine	700
139	lwarp-lineno.sty	701
231	lineno	701
140	lwarp-lips.sty	704
232	lips	704
141	lwarp-listings.sty	704
233	listings	704
142	lwarp-longtable.sty	708

234	longtable	708
143	lwarp-lscape.sty	710
235	lscape	710
144	lwarp-ltablex.sty	710
236	ltablex	710
145	lwarp-ltcaption.sty	711
237	ltcaption	711
146	lwarp-ltxgrid.sty	711
238	ltxgrid	711
147	lwarp-ltxtable.sty	712
239	ltxtable	712
148	lwarp-lua-check-hyphen.sty	712
240	lua-check-hyphen	712
149	lwarp-luacolor.sty	712
241	luacolor	712
150	lwarp-luatodonotes.sty	713
242	luatodonotes	713
151	lwarp-magaz.sty	715
243	magaz	715

152	lwarp-marginfit.sty	715
244	marginfit	715
153	lwarp-marginfix.sty	716
245	marginfix	716
154	lwarp-marginnote.sty	716
246	marginnote	716
155	lwarp-mcaption.sty	717
247	mcaption	717
156	lwarp-mdframed.sty	717
248	mdframed	717
248.1	Limitations	717
248.2	Package loading	718
248.3	Patches	718
248.4	Initial setup	719
248.5	Color and length HTML conversion	719
248.6	Environment encapsulation	720
248.7	Mdframed environment	721
248.8	Titles and subtitles	722
248.9	New environments	725
157	lwarp-memhfixc.sty	727
249	memhfixc	727
158	lwarp-metalogo.sty	728
250	metalogo	728
159	lwarp-mhchem.sty	728
251	mhchem	728

160	lwarp-microtype.sty	730
252	microtype	730
161	lwarp-midfloat.sty	731
253	midfloat	731
162	lwarp-midpage.sty	732
254	midpage	732
163	lwarp-morefloats.sty	732
255	morefloats	732
164	lwarp-moreverb.sty	732
256	moreverb	732
165	lwarp-morewrites.sty	734
257	morewrites	734
166	lwarp-mparhack.sty	734
258	mparhack	734
167	lwarp-multicol.sty	734
259	multicol	734
168	lwarp-multirow.sty	735
260	multirow	735
260.1	Multirow	736
260.2	Combined multicolumn and multirow	737
169	lwarp-multitoc.sty	739

261	multitoc	739
170	lwarp-nameref.sty	739
262	nameref	739
171	lwarp-natbib.sty	739
263	natbib	739
172	lwarp-needspace.sty	740
264	needspace	740
173	lwarp-newclude.sty	741
265	newclude	741
174	lwarp-newunicodechar.sty	741
266	newunicodechar	741
175	lwarp-nextpage.sty	741
267	nextpage	741
176	lwarp-nicefrac.sty	742
268	nicefrac	742
177	lwarp-nonfloat.sty	742
269	nonfloat	742
178	lwarp-nonumonpart.sty	743
270	nonumonpart	743

179	lwarp-nopageno.sty	743
271	nopageno	743
180	lwarp-nowidow.sty	743
272	nowidow	743
181	lwarp-ntheorem.sty	744
273	ntheorem	744
273.1	Limitations	744
273.2	Options	744
273.3	Remembering the theorem style	745
273.4	HTML cross-referencing	748
273.5	\newtheoremstyle	748
273.6	Standard styles	749
273.7	Additional objects	750
273.8	Renewed standard configuration	751
273.9	amsthm option	752
273.10	Ending a theorem	755
273.11	\NoEndMark	755
273.12	List-of	755
273.13	Symbols	756
273.14	Cross-referencing	756
182	lwarp-overpic.sty	756
274	overpic	756
183	lwarp-pagegrid.sty	757
275	pagegrid	757
184	lwarp-pagenote.sty	757
276	pagenote	757
185	lwarp-pagesel.sty	758
277	pagesel	758

186	lwarp-paralist.sty	758
278	paralist	758
187	lwarp-parnotes.sty	759
279	parnotes	759
188	lwarp-parskip.sty	760
280	parskip	760
189	lwarp-pbox.sty	760
281	pbox	760
190	lwarp-pdfscape.sty	761
282	pdfscape	761
191	lwarp-pdfrender.sty	761
283	pdfrender	761
192	lwarp-pdfsync.sty	761
284	pdfsync	761
193	lwarp-pfnote.sty	762
285	pfnote	762
194	lwarp-phfqit.sty	762
286	phfqit	762
195	lwarp-placeins.sty	762

287	placeins	762
196	lwarp-prelim2e.sty	763
288	prelim2e	763
197	lwarp-prettyref.sty	763
289	prettyref	763
198	lwarp-preview.sty	763
290	preview	763
199	lwarp-quotchap.sty	764
291	quotchap	764
200	lwarp-quoting.sty	765
292	quoting	765
201	lwarp-ragged2e.sty	766
293	ragged2e	766
202	lwarp-realscripts.sty	766
294	realscripts	766
203	lwarp-register.sty	767
295	register	767
204	lwarp-relsize.sty	768
296	relsize	768

205	lwarp-resizegather.sty	769
297	resizegather	769
206	lwarp-romanbar.sty	770
298	romanbar	770
207	lwarp-romanbarpagenumber.sty	770
299	romanbarpagenumber	770
208	lwarp-rotating.sty	770
300	rotating	770
209	lwarp-rotfloat.sty	771
301	rotfloat	771
210	lwarp-savetrees.sty	772
302	savetrees	772
211	lwarp-scalefnt.sty	772
303	scalefnt	772
212	lwarp-schemata.sty	772
304	schemata	772
213	lwarp-scrextend.sty	773
305	scrextend	773
214	lwarp-scrhack.sty	776
306	scrhack	776

215	lwarp-scrlayer.sty	776
307	scrlayer	776
216	lwarp-scrlayer-notecolumn.sty	778
308	scrlayer-notecolumn	778
217	lwarp-scrlayer-scrpage.sty	778
309	scrlayer-scrpage	778
218	lwarp-section.sty	779
310	section	779
219	lwarp-sectionbreak.sty	780
311	sectionbreak	780
220	lwarp-sectsty.sty	781
312	sectsty	781
221	lwarp-setspace.sty	781
313	setspace	781
222	lwarp-shadow.sty	783
314	shadow	783
223	lwarp-showidx.sty	783
315	showidx	783
224	lwarp-showkeys.sty	783
316	showkeys	783

225	lwarp-sidecap.sty	784
317	sidecap	784
226	lwarp-sidenotes.sty	785
318	sidenotes	785
227	lwarp-siunitx.sty	786
319	siunitx	786
228	lwarp-soul.sty	792
320	soul	792
229	lwarp-soulpos.sty	794
321	soulpos	794
230	lwarp-soulutf8.sty	794
322	soulutf8	794
231	lwarp-stabular.sty	795
323	stabular	795
232	lwarp-stfloats.sty	795
324	stfloats	795
233	lwarp-subfig.sty	796
325	subfig	796
234	lwarp-subfigure.sty	801
326	subfigure	801

235	lwarp-supertabular.sty	801
327	supertabular	801
236	lwarp-syntonly.sty	803
328	syntonly	803
237	lwarp-tlenc.sty	803
329	tlenc	803
238	lwarp-tabls.sty	804
330	tabls	804
239	lwarp-tabularx.sty	804
331	tabularx	804
240	lwarp-tabulary.sty	804
332	tabulary	804
241	lwarp-textarea.sty	805
333	textarea	805
242	lwarp-textcomp.sty	805
334	textcomp	805
334.1	Limitations	806
334.2	Package loading	806
334.3	Remembering original defintions	806
334.4	HTML symbols	807
334.4.1	pdf \TeX symbols	807
334.4.2	X \mathbb{Y} \TeX and Lua \mathbb{Y} \TeX symbols	807
334.5	HTML dicritics	808
334.6	Inside a lateximage	808

243	lwarp-textfit.sty	809
335	textfit	809
244	lwarp-textpos.sty	810
336	textpos	810
245	lwarp-theorem.sty	810
337	theorem	810
337.1	Remembering the theorem style	811
337.2	CSS patches	812
246	lwarp-threeparttable.sty	815
338	threeparttable	815
247	lwarp-tikz.sty	816
339	tikz	816
248	lwarp-titleps.sty	817
340	titleps	817
249	lwarp-titleref.sty	820
341	titleref	820
250	lwarp-titlesec.sty	821
342	titlesec	821
251	lwarp-titletoc.sty	823
343	titletoc	823
252	lwarp-titling.sty	825

344	titling	825
-----	---------	-----

253	lwarp-tocbasic.sty	830
-----	--------------------	-----

345	tocbasic	830
-----	----------	-----

254	lwarp-tocbibind.sty	830
-----	---------------------	-----

346	tocbibind	830
-----	-----------	-----

255	lwarp-tocenter.sty	832
-----	--------------------	-----

347	tocenter	832
-----	----------	-----

256	lwarp-tocloft.sty	833
-----	-------------------	-----

348	tocloft	833
-----	---------	-----

257	lwarp-tocstyle.sty	839
-----	--------------------	-----

349	tocstyle	839
-----	----------	-----

258	lwarp-todo.sty	840
-----	----------------	-----

350	todo	840
-----	------	-----

259	lwarp-todonotes.sty	841
-----	---------------------	-----

351	todonotes	841
-----	-----------	-----

260	lwarp-transparent.sty	843
-----	-----------------------	-----

352	transparent	843
-----	-------------	-----

261	lwarp-trimclip.sty	843
-----	--------------------	-----

353	trimclip	843
-----	----------	-----

262	lwarp-trivfloat.sty	844
354	trivfloat	844
354.1	Combining \newfloat, \trivfloat, and algorithmicx	845
263	lwarp-turnthepage.sty	845
355	turnthepage	845
264	lwarp-typearea.sty	845
356	typearea	845
265	lwarp-ulem.sty	846
357	ulem	846
266	lwarp-underscore.sty	848
358	underscore	848
267	lwarp-upref.sty	848
359	upref	848
268	lwarp-url.sty	849
360	url	849
269	lwarp-verse.sty	849
361	verse	849
270	lwarp-vertbars.sty	851
362	vertbars	851
271	lwarp-vmargin.sty	852

363	vmargin	852
-----	----------------	------------

272	lwarp-vowel.sty	852
-----	------------------------	------------

364	vowel	852
-----	--------------	------------

273	lwarp-vwcol.sty	853
-----	------------------------	------------

365	vwcol	853
-----	--------------	------------

274	lwarp-wallpaper.sty	855
-----	----------------------------	------------

366	wallpaper	855
-----	------------------	------------

275	lwarp-wasysym.sty	856
-----	--------------------------	------------

367	wasysym	856
-----	----------------	------------

276	lwarp-watermark.sty	856
-----	----------------------------	------------

368	watermark	856
-----	------------------	------------

277	lwarp-wrapfig.sty	856
-----	--------------------------	------------

369	wrapfig	856
-----	----------------	------------

278	lwarp-xcolor.sty	858
-----	-------------------------	------------

370	xcolor	858
-----	---------------	------------

370.1	Limitations	858
-------	-----------------------	-----

370.2	Xcolor definitions: location and timing	858
-------	---	-----

370.3	Package loading	861
-------	---------------------------	-----

370.4	Remembering and restoring original definitions	861
-------	--	-----

370.5	HTML color style	861
-------	----------------------------	-----

370.6	HTML border	862
-------	-----------------------	-----

370.7	High-level macros	863
-------	-----------------------------	-----

370.8	Row colors	867
-------	----------------------	-----

279	lwarp-xellipsis.sty	869
-----	----------------------------	------------

371	xellipsis	869
280	lwarp-xfrac.sty	870
372	xfrac	870
281	lwarp-xltabular.sty	873
373	xltabular	873
282	lwarp-xltxtra.sty	873
374	xltxtra	873
283	lwarp-xmpinl.sty	874
375	xmpinl	874
284	lwarp-xpiano.sty	874
376	xpiano	874
285	lwarp-xtab.sty	875
377	xtab	875
286	lwarp-xurl.sty	877
378	xurl	877
287	lwarp-xy.sty	877
379	xy	877
288	lwarp-zwpgelayout.sty	878
380	zwpgelayout	878

289 lwarp-patch-komascript.sty 879

381 patch-komascript 879

290 lwarp-patch-memoir.sty 881

382 patch-memoir 881

382.1 Packages	882
382.2 Preliminary setup	883
382.3 Laying out the page	883
382.4 Text and fonts	886
382.5 Titles	887
382.6 Abstracts	887
382.7 Document divisions	887
382.8 Pagination and headers	890
382.9 Paragraphs and lists	891
382.10 Contents lists	892
382.11 Floats and captions	897
382.12 Page notes	900
382.13 Decorative text	902
382.14 Poetry	902
382.15 Boxes, verbatims and files	903
382.16 Cross referencing	903
382.17 Back matter	904
382.18 Miscellaneous	904
382.19 Ccaption emulation	905
382.20 Final patchwork	907

Change History and Index 908

List of Figures

1	tutorial.tex listing	80
---	----------------------	----

List of Tables

1	\LaTeX -HTML generation — lwarp package — Supported features	63
2	Required software programs	72
3	Files created along with the print version	82
4	Package options	96
5	HTML settings	98
6	Section HTML headings for word-processor conversion	148
7	Section depths and HTML headings	160
8	Tabular baseline	378
9	Tabular HTML column conversions	378
10	Cross-referencing data structures	425
11	Float data structures	435
12	AMStm package — CSS styling of theorems and proofs	561
13	Ntheorem package — CSS styling of theorems and proofs	744
14	Theorem package — CSS styling of theorems and proofs	811

2 Updates

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

For a detailed list of the most recent changes, see the end of the Change History on page 929.

v0.57:

MathJax
math macros

dynamic math

 new name
lateximage alt tags

misc. fixes

packages

- Added support for MathJax equations with `\footnote`, `\footnotemark`.
- Added `\StartDefiningMath` and `\StopDefiningMath` for use when defining macros in the preamble which contain `$`. See section 9.6.6.
- Added `\StartDynamicMath` and `\StopDynamicMath` to delimit math expressions which depend on a variable condition such as a counter. Such expressions will not be hashed for reuse, and will be converted to svg math images even when MathJax is enabled. See section 9.6.7.
- Renamed `\EndDefiningTabulars` to `\StopDefiningTabulars`.
- Improved localization for `lateximage` HTML alt tags. For svg math images, the alt tag under some conditions will be set to `\mathimagenname`, which defaults to “math image”. For packages, the alt tag is set using the package name followed by `\packagediagramname`, which defaults to “diagram”. Ex:

$$(-xy- \text{diagram})$$
 See section 8.4.
- Fix: Restoring various macros in a `lateximage` and similar situations.
- Fix: `\href` text catcodes.
- Fix: `\subref` text.
- Fixes: Colored `\rule` and `\boxframe`.
- **float**, **rotfloat**: Adds support for float styles ruled and boxed.
- **float**: Fix: Do not create `\l@<type>` until `\listof` is used.
- **marginnote**: Fix: Long optional argument.
- **ellipsis**: Adds `\midwordellipsis`.
- **breakurl**: Fix for text catcodes.
- Added **algorithm2e**, **register**, **ltablex**, **xltabular**, **xellipsis**, **trimclip**, **errata**, **vowel**, **xpiano**.
- Prevents **glossary**.
- Verified works as-is with **gauss**, **phonrule**, **piano**, **Slunits**, **tikzcodeblocks**.

v0.56:**lwarpmk**

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

tabular

- Added support for `-shell-escape`. See section 8.1.
- Added support for **array** w and W columns.
- Fix: `\multicolumn` parameter handling.
- Added support for double `\hlines`, `\midrules`, and vertical rules.
- Added support for **arydshln** dashed lines with HTML tabular, but reverts to plain rules for lateximage and SVG math array.

misc. fixes

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

packages

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

v0.55:**misc fixes**

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

packages

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

v0.54:**lwarpmk**

- **lwarpmk** `limages` checks for the presence of the HTML version of the document and valid image references before attempting to create the lateximages.

- **lwarpmk**: Improved error message if configuration file does not exist.

BibTeX

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

polyglossia**macros in section names**

- Added documentation regarding **polyglossia**. See section 9.13.4.
- Added documentation regarding the use of macros in section names. See section 9.1.

document encoding

⚠ New and revised encoding options

- Renamed and added package options:

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

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

For an existing document, recompile the print version with **pdflatex**, **xelatex**, or **lualatex** a single time, to readjust the `lwarpmk.conf` configuration file, before recompiling the HTML version.

⚠ Reset the configuration

floats with `\centering`, etc.

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

misc. fixes

- **tikz**: `\pgfpicture`, `fit`, `align`, `font`.
- **ragged2e**: `\centering` etc.
- **hyperref**: `\hypertarget` was creating duplicate of `\label`.
- **hyperref**: Active chars inside `\hyperref`, `\hyperlink`.
- **hyperref**: `\ref` inside `\hyperlink` caused a nested HTML link.
- **glossaries**: Fix when not using **babel** or **polyglossia**.
- **textcomp**: `\textperthousand`.
- **L^AT_EX** core verse environment: line spacing.
- Removed `\citetitle`, adjusted `\attribution`.
- **memoir**: Minor update for v3.7g.
- Added **inputenx**, **bibunits**, **chnpage**, **forest**, **magaz**, **gridset**.
- Prevents loading **ae**, **aecc**, **t1enc**, and **wasysym**.



packages

v0.53:

lwarpmk

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

v0.52:

documentation

- Improved install instructions regarding `lwarp_baseline_marker.png`.
- Added documentation regarding footnotes in section headings, and footnotes with `\VerbatimFootnotes` from **fancybox**, **fancyvrb**. See section 9.4.4.
- Added documentation regarding font selection when using \LaTeX or \LuaTeX with **fontspec** and traditional font packages. See section 8.2.

SVG math

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

MathJax

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

footnotes

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

misc. fixes

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

packages

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

v0.51:

documentation

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

SVG math

- Improved svg math baseline.

- Improved svg math font and color.
 - Faster svg math rendering.
 - Improved support for display math containing complicated math objects, such as **tikz-cd**. See section 9.6.8.
 - Fix: `\addcontentsline` inside svg math.
 - Fix: SVG math containing an embedded `lateximage`.
 - MathJax now handles `\ensuremath` in expressions.
 - Fix: Added `alignat` environment.
 - Fix: **afterpackage** no longer required, which conflicted with `scrfile`.
 - Fix: **titling** `\thanks` mark.
 - Fix: **fancybox** improvements.
 - Fix: **tikz** `\tikz` macro. (Previously only the `tikzpicture` environment worked.)
 - Fix: **tikz** with optional argument.
 - Added **mhchem**, **chemfig**, **chemformula**, **chemmacros**, **chemnum**, **chemgreek**, **epstopdf-base**, **grid**, **ltxgrid**.
- v0.50:**
- svg math and other `lateximages` now are converted to svg using parallel background tasks, utilizing all available CPU cores.
 - Inline svg math image file names now are MD5 hashes made from their source \TeX code. Identical inline math expressions, such as multiple instance of $\$x\$$, now share a single image file. This reduces the number of images to store, transmit, process, and display. Each image file is only converted to svg a single time, and reused if it already exists. Display math and other forms of svg image such as `picture` and `Tikz` still use individual image files which are recreated each time *lwarpmk images* is run.
 - Fixes: SVG math and/or `\underline` in a sectioning file name.
 - Improved svg display math and tags.
 - Improved svg math and **siunitx** alt tags.
 - Improved **siunitx** units.
 - Fix: `\ensuremath` with MathJax now creates a `lateximage`.
 - Fix: `\centering`, etc. in svg math, `lateximage`, `Tikz`.
 - Fix: Made various macros robust, additionally fixing **authblk**.
 - Fix: **ntheorem** if neither standard nor `amsthm` selected.
 - Fix: **listings**: Improved column alignment.
 - Fix: Load **fontspec** if necessary.
 - Added **xy**, **epstopdf**, **diagbox**, **pbox**, **bytefield**, **axodraw2**, **phfqit**, **schemata**, **dblfloatfix**, **nonfloat**, **morefloats**.

v0.49:

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

v0.48:

- documentation**
 - Added some documentation regarding converting an existing document. See section 7.
- cleveref**
 - Updated compatibility for new **cleveref** v0.21.
- tabular**
 - Fix: Ignores optional tabular column arguments.
- minor updates**
 - Added \leftline, \centerline, \rightline.
 - Lists have improved font control via \makelabel.
 - Print-mode lateximage now boxed to the natural width of its multiline contents.
 - abstract now allows an optional name, as required by some classes.
- math**
 - Fix: Improved spacing, \mbox, and font sizes with svg math, 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, \crr, \noalign, for **IEEEtran** class.
 - Fix: **xfrac** neutralized BlockClass and others.
 - Fix: **todonotes** and **luatodonotes**: Improved \todotoc.
- 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:

documentation

memoir

cross-references

math

bibliography

misc. fixes

packages

- Improved **MikTeX** install instructions.
- Improved graphics and **epstopdf** instructions.
- Updates to the **Introduction**.
- Added **memoir**, **memhfixc**. See section 9.12.
- Fix: Now allows underscores in labels.
- Fix: `_` and `\<blank>` in section/file names.
- Fix: Now allows MATHJAX inside `tabbing`.
- Fix: Bibliography `\em` names.
- Added **cite**, **natbib**, **backref**. (Also works as-is with **biblatex**.)
- 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.
- Added **subfigure**, **prettyref**, **hanging**, **midpage**, **flafter**, **fltrace**, **changebar**, **endfloat**, **continue**, **fwlw**, **turnthepage**, **footnpag**, **pagesel**, **textfit**, **titleref**.

v0.44:

koma-script

HTML title and author

encodings

- Added **koma-script** classes (except **scrلتtr2**, **scrjura**).
- Added **scrxextend**, **scrlayer**, **scrlayer-notecolumn**, **scrlayer-scrpage**, **scrhack**, **tocstyle**, **tocbasic**.
- 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.
- 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**, **ifoddpaper**.
 - Added **preview**, **atbegshi**, **watermark**.
 - Improved **tocloft** `\newlistof` and `\newlistentry`.

v0.43:

- footnotes**
 - Docs: Reorganized HTML customization, added an HTML settings table. See section 8.4.
 - 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 T_EX!**
 - Added T_EX development support page, **Supporting T_EX development**.
- word-processor conversion**
 - Improved assistance for word-processor conversions when boolean `FormatWP` is set true. See section 11.
 - The boolean `FormatWordProcessor` has been renamed `FormatWP`.
 - The boolean `HTMLMarkFloats` has been renamed `WPMarkFloats`.

⚠ name change

⚠ name change

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

tabbing

overpic

math

symbols

load order

- Add: tabbing environment.
- Add: **overpic** package. See section 274.
- Fix: Text copy/paste of \mathcal{AMS} math environment numbers and names.
- Improved `\ensuremath`.
- **MATHJAX** with **siunitx**: Updated script and documentation.
- **textcomp**: Improved `\interrobangdown`.
- **realscripts**: Fix for subscripts in a `lateximage`.
- **morewrites**: Enforces loading before **lwarp**.

v0.41:

tabular

 new syntax

- Added tabular vertical rules, subject to some limitations. See the rules section of section 9.9.
- 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 260.2 on page 737.
- Fix: Unfinished tabular rows are automatically filled.
- Fix for tabular column specifiers while using **babel-french**. (`\NoAutoSpacing` is activated then nullified inside the tabular, due to a conflict with the tabular column parsing code.)

v0.40:

graphics, graphicx

`\includegraphics path`

 image file extensions

bigdelim

symbols

fixes

- **graphics** and **graphicx** have been moved from the **lwarp** core, and are only loaded if requested with `\usepackage`.
- Improved **graphics** `\graphicspath` support. Multiple image directories may now be used. Refer to **.pdf files without a file extension** to allow the HTML version to use a `.svg`, `.png`, `.jpg`, or `.gif` version instead. See section 9.7.
- **grffile** is now directly supported instead of emulated.
- Fix for **bigdelim**, and improved documentation. See section 121.
- Improved \TeX and **textcomp** symbols.
- Fix for \TeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for **xltxtra** with $X_{\mathbb{E}}\TeX$.

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

margins
columns
footnotes
tabular
sectioning

v0.39:

- Improved the titlepage HTML code, `\thanks` notes, and `\maketitle`. **titling** is no longer required, but is still supported. The `\published` and `\subtitle` fields are no longer provided, but `\AddSubTitlePublished` replicates them using **titling**. See section 62.8. **authblk** is added, and should be loaded before **titling**. See section 62.
- `\multirow` now supports the new optional `vpos` argument.
- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 260.2.
- Tabular special cases:
 - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 9.9.
 - Added `\ResumeTabular` for use when a tabular environment is defined inside another environment. See section 9.9.
- Added **supertabular**, **xtab**, **bigstrut**, **bigdelim**.
- Added **fullwidth**.
- Added **addlines**, **anysize**, **a4**, **a4wide**, **a5comb**, **textarea**, **zwpagelayout**, **typearea**, **ebook**.

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

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

v0.38:

- Added `lwarpmk print1` and `lwarpmk html1` actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.
- Added `\ForceHTMLPage` and `\ForceHTMLTOC` to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 9.5.2.
- Modified the tutorial to use the new `\ForceHTMLPage` and `\ForceHTMLTOC` macros.
- Added **appendix**, **tocbibind**, **fncychap**, **fix2col**.
- Added **relsize**, **scalefnt**.

forced single-pass compile

starred sections

updated tutorial

packages
font size

page numbering
front & back matter

- Added **realscripts**, **metalogo**, **xltxtra**.
- Added **grffile**, **romanbar**.
- Added **arabicfront**, **chappg**, **nonumonpart**, **nopageno**, **romanbarpagenumber**.
- Docs: Improved description of the use of front/back matter. See section 9.5.
- Fix: **color** requests **xcolor**.
- Fix: `\part` for **article** class.

v0.37:

`\include` for HTML
latexmk

accents and symbols
babel-french

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

v0.36:

boxes and frames

babel-french

footnotes

siunitx

- Recorganized the documentation section regarding special cases and limitations. (Section 9)
- Improved source formatting.
- `\fbox` and related now use `\fboxsep` and `\fboxrule`.
- `\makebox` and `\framebox` now use width and position.
- `\fcolorbox` and related now work inside a `lateximage`.
- **babel-french**: Improvements for French variants, load order, footnotes, ellipses.
- Improved footnote numbering. `lateximage` footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with **MATHJAX**.
- Improved **siunitx** units.
- Fix for filenames while using **MATHJAX**.
- Fix for `\rule` when **xcolor** is not loaded.
- Added **transparent**, **upref**.

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

v0.34:

⚠ Optional arguments

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

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

instead of:

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

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

⚠ Optional arguments

spans with minipages

framing minipages

lateximage, SVG math,
tabular

eqnarray

verbatim packages

framing packages

list packages

babel-french

- Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:
`\InlineClass[style]{class}{text}`
- 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 9.3.3. `\fboxBlock` is added to frame minipages, tables, and lists with full HTML formatting but no longer inline, and behaves as `\fbox` for print output. The `fminipage` environment is added for framed minipages, as an environment with full HTML formatting, and draws a framed minipage in print output. See section 9.3.5. `\fbox` and minipages now often work in SVG math and `lateximages`. `MATHJAX` supports `\fbox`, but not `\fboxBlock` nor `fminipage`.
- Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, **booktabs**.
- Improved font control for `lateximages` 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 `\StopDefiningTabulars` (previously called `\EndDefiningTabulars`). See section 40. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the & character inside the definition. This may include the use inside conditional expressions.

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

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

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

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

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

v0.30:

lwarp-newproject

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

To modify existing documents, remove from the document source:

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

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

HTML setup changes.

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

Old Macro	New Package Option
<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>xindyLanguage</code> <code>xindyStyle</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={},
  xindyLanguage=english
]{lwarp}
```

- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
 1. Make the changes shown above.
 2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
 3. The old file `lwarp_html.xdy` may be deleted.
- The new **lwarp** package option `optnxindyStyle` may be used to tell **lwarpmk** to use a custom `.xdy` file instead of `lwarp.xdy`. See section 8.13.
- Improvements in index processing:
 - **xindy**'s language is now used for index processing as well as glossary.
 - Print mode without **latexmk** now uses **xindy** instead of **makeindex**.
 - **texindy**/**xindy** usage depends on **pdflatex** vs **xelatex**, **lualatex**.
 - For **pdflatex** and **texindy**, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

v0.29:

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

v0.28:

- `\HTMLAuthor {<name>}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- 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 the boolean `FormatWordProcessor` is true. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames. Name changed to `WPMarkFloats` as of v0.42.
- Updated for the new MATHJAX CDN repository.
- Adds **tabulary**.
- Supports the options syntax for **graphics**.
- Improved index references, now pointing exactly to their target.
- Adds **glossaries**. `lwarpmk` is modified to add `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. \TeX itself is still used, allowing a wider range of \TeX trickery to be understood. 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, specialized math-mode typesetting for various fields of science and engineering, and anything generating plain-text output. Blocks of PDF output may be automatically converted to SVG images while using the same font and spacing as the original print document, directly supporting `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: perceived importance, popularity lists, recent CTAN updates, CTAN topics, mention in other packages, support by other HTML conversion methods, and from sample documents taken from public archives. These include some “obsolete” packages as well.¹

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

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

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

SVG images may be used for math, and are also used for `picture`, `Tikz`, and similar environments. The SVG format has better browser and e-book support than MathML (as of this writing), while still allowing for high-quality display and printing of images (again, subject to potentially bug-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 . An MD5 hash is used to combine multiple instances of the same inline math expression into a single image file, which then needs to be converted to SVG only a single time.

The MATHJAX JavaScript display engine may be selected for math display instead of using SVG images. Subject to browser support and Internet access, MATHJAX allows an HTML page to display math without relying on a large number of external image files.⁴ **lwarp** maintains \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-package.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, except that duplicate inline math expressions are combined into a single file according to the MD5 hash function of its contents. A common scientific paper can easily include several thousand files, and in one case the MD5 hash cut the number of files in half and the rendering time by 30%.

3.1 Supported packages and features

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

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

Category	Status and supported features.
Engines:	pdf \LaTeX , Xe \LaTeX , Lua \LaTeX
Classes:	article , book , report , scrartcl , scrbook , scrreprt , memoir .
Koma-script:	scrextend , scrhack , scrlayer . Others as listed below.
Memoir:	memhfixc
Page layout:	a4 , a4wide , a5comb , addlines , anysize , atbegshi , blowup , clrdblpg , continue , draftwatermark , ebook , everyshi , fancyhdr , fwlw , geometry , grid , gridset , ltxgrid , pagegrid , pagesel , preview , scrlayer-scrpage , textarea , titleps , tocenter , turnthepage , typearea , vmargin , watermark , zwpagelayout .
Sectioning:	Adds <code>FileDepth</code> for splitting the HTML output. Files may be numbered sequentially or named according to section name. Common short words and punctuation are removed from the file-names. anonchap , fncychap , quotchap , section , sectionbreak , sectsty , titlesec .
Table of contents, figures, tables:	Supported, with hyperlinks. multitoc , shorttoc , titletoc , tocbasic , tocbibind , tocloft , tocstyle .
Title page:	<code>\maketitle</code> , <code>titlepage</code> , authblk , titling .
Front & back matter:	abstract , appendix .
Indexing:	texindy is used, with hyperlinks. idxlayout .
Glossary:	glossaries and xindy are used.
Bibliography:	babelbib , backref , biblatex , bibunits , chapterbib , cite , hypernat , natbib .

lwarp Supported Functions — continued

Category	Status
Cross-references:	bookmark, breakurl, cleveref, fancyref, hypdestopt, hyperref, prettyref, titleref, url, varioref, xurl.
Languages:	babel, polyglossia.
Margin notes:	marginfit, marginfix, scrlayer-notecolumn.
Footnotes:	Adds FootnoteDepth to print footnotes at section breaks. endheads, endnotes, footmisc, footnote, footnpag, marginnote, nccfoots, pagenote, parnotes, sidenote.
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, amsthm, ntheorem, theorem.
Additional math:	Math fonts via SVG images, resizegather, xy. Tested to work as-is: amscd, bm, braket, delarray, guass, nicematrix, pb-diagram, tikz-cd, etc.
Display math with $\backslash displaymathother$:	Complicated math objects in display math, such as tikz-cd, etc.
Units and fractions:	nicefrac, siunitx, units, xfrac. Tested to work as-is: Slunits.
Floats:	Appear where declared. capt-of, caption, cutwin, dblfloatfix, endfloat, fix2col, flafter, float, floatflt, floatrow, fltrace, hypcap, keyfloat, morefloats, newfloat, nonfloat, placeins, rotfloat, stfloats, subcaption, subfig, subfigure, subfloat, trivfloat, wrapfig.
Tabular:	tabular environment, array, arydshln, bigdelim, booktabs, colortbl, diagbox, longtable, ltablex, ltxtable, multirow, supertabular, tabularx, tabulary, threeparttable, xtabular, xtab.

lwarp Supported Functions — continued

Category	Status
Graphics:	graphics and graphicx . <code>\includegraphics</code> supports width, height, origin, angle, and scale tags, and adds <code>class</code> . References to PDF files are changed to <code>svg</code> , other image types are accepted as well. <code>\rotatebox</code> and <code>\scalebox</code> are supported as well as HTML can handle. rotating is emulated but all objects are unrotated. picture , tikz , and xy are converted to an <code>svg</code> image. epstopdf , figsize , grffile , overpic . Tested to work as-is: tikz-3dplot .
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, enumerate , enumitem , hang , paralist .
Environments:	Standard \LaTeX environments.
<code>minipage</code> , <code>\parbox</code> :	Some HTML5-imposed limitations. Nested minipages are supported. pbox .
Quotations:	csquotes , epigraph , quoting , verse .
Verbatim:	fancyvrb , moreverb , shortvrb , verbatim .
Frames:	boxedminipage2e , fancybox , framed , mdframed , shadow , vertbars .
Multi-columns:	adjmulticol , multicol , vwcol .
Margins:	fullwidth , hanging , midpage .
Line numbering:	fnlineno , lineno .
Acronyms:	acro , acronym .
Todo notes:	changebar , easy-todo , errata , fixme , fixmetodonotes , todo , todonotes .

lwarp Supported Functions — continued

Category	Status
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are only supported in some cases. <code>cancel</code>, <code>ellipsis</code>, <code>hyphenat</code>, <code>lettrine</code>, <code>lips</code>, <code>lua-check-hyphen</code>, <code>luacolor</code>, <code>magaz</code>, <code>pdfrender</code>, <code>realscripts</code>, <code>reysize</code>, <code>scaleftnt</code>, <code>soul</code>, <code>soulpos</code>, <code>soulutf8</code>, <code>textfit</code>, <code>trimclip</code>, <code>ulem</code>, <code>underscore</code>, <code>xellipsis</code>.
Ordinals:	<code>engord</code>, <code>fmtcount</code>, <code>nth</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> .
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>chemgreek</code>, <code>textalpha</code>, <code>textcomp</code>, <code>textgreek</code>.
Science and engineering:	<code>algorithm2e</code>, <code>algorithmicx</code>, <code>axodraw2</code>, <code>bytefield</code>, <code>chemfig</code>, <code>chemformula</code>, <code>chemgreek</code>, <code>chemmacros</code>, <code>chemnum</code>, <code>listings</code>, <code>mhchem</code>, <code>phfqit</code>, <code>register</code>. Tested to work as-is: <code>blochsphere</code>, <code>bodegraph</code>, <code>bohr</code>, <code>circuitikz</code>, <code>elements</code>, <code>fast-diagram</code>, <code>hepnicenames</code>, <code>heppennames</code>, <code>linop</code>, <code>pgfgantt</code>, <code>physics</code>, <code>simpler-wick</code>, <code>slashed</code>, <code>structmech</code>, <code>tikzcodeblocks</code>.
Liberal arts and humanities:	<code>forest</code>, <code>schemata</code>, <code>vowel</code>, <code>xpiano</code>. Tested to work as-is: <code>phonrule</code>, <code>piano</code>, <code>tikz-dependency</code>.

lwarp Supported Functions — continued

Category	Status
Working as-is:	Various utility, calculation, file, and text-only packages, such as calc , fileerr , somedefs , trace , xspace . Also, any math-only packages, including specialized typesetting for various fields of science and engineering.

4 Alternatives

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

4.1 Internet class

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

4.2 TeX4ht

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

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

4.3 Translators

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

Prog `Hevea` **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 be used to directly insert \LaTeX math into HTML:

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

4.4 AsciiDoc and AsciiDoctor

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

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

4.4.1 AsciiDoctor-LaTeX

The AsciiDoctor-LaTeX project is developing additional \LaTeX -related features.

AsciiDoctor-LateX:

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

The lwarp package: This package.

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

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

xindy: The **xindy** program is used by **lwarp** to create indexes. On a $\text{MiK}\text{\TeX}$ 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 $\text{MiK}\text{\TeX}$ 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 $\text{MACOS}^{\text{\textregistered}}$, see <https://brew.sh/>, install **Homebrew**, then

```
Enter  $\Rightarrow$  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 **MiK_TE_X Package Manager (Admin)**.
2. To update **lwarp**, use **MiK_TE_X 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 **MiK_TE_X Settings (Admin)** and click on the button called Refresh FNDB.

CTAN .dtx and .ins files: Another form of T_EX package is .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 `lwarp_baseline_marker.png` to the same place as the .sty files.
9. Copy the documentation `lwarp.pdf` to a source directory in the local tree, such as:


```
<texlocal>/doc/local/lwarp
```
10. Renew the cache:


```
Enter ⇒ mktexlsr
```

 — or —


```
Enter ⇒ texhash
```

 Or, for WINDOWS MiK_TE_X, start the program called **MiKTeX Settings (Admin)** and click on the button called Refresh FNDB.
11. See section 5.2.1 to generate your local copy of **lwarpmk**.
12. Once the local version of `lwarpmk.lua` is installed, it may be made available system-wide as per section 5.2.

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

Just testing!

5.2 Installing the lwarpmk utility

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

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

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

Otherwise, continue with the following:

3. Locate the file `lwarpmk.lua`, which should be in the `scripts` directory of the `tds` tree. On a 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 72.

```
Enter ⇒ luatex --version
Enter ⇒ xindy --version
Enter ⇒ latexmk --version
Enter ⇒ perl --version
Enter ⇒ pdfcrop --version
Enter ⇒ pdftotext -v
Enter ⇒ pdfseparate --version
Enter ⇒ pdftocairo -v
```

To install xindy, latexmk, and pdfcrop:

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

```
http://ctan.org/pkg/xindy
http://ctan.org/pkg/latexmk
http://ctan.org/pkg/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 page 72.
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.

[Need help?](#)

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

6.1 Starting a new project

1. Create a new project directory called `tutorial`.
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}
\fbbox{\textcolor{blue!50!green}{Text in a figure.}}
\caption{A figure with text\label{fig:withtext}}
\end{center}\end{figure}

\section{Some math}

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


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

\end{document}

```

Table 3: Files created along with the print version

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

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

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

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

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

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

6.2 Compiling the print version with lwarpmk

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

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

1. Re-compile the print version:

Enter ⇒ `lwarpmk print`

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

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

Enter ⇒ `lwarpmk print`

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

4. Force a recompile to occur.

Enter ⇒ `lwarpmk again`

Enter ⇒ `lwarpmk print`

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

5. Process the index.⁶⁷

Enter ⇒ `lwarpmk printindex`

6. Recompile again to include the index.

Enter ⇒ `lwarpmk print`

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

Enter ⇒ `lwarpmk print1`

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

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 9.5.10.

⁷Also see section 9.5.11 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 \Rightarrow `lwarpmk html`

- (a) **lwarpmk** uses \LaTeX to process `tutorial_html.tex` to create `tutorial_html.pdf`.
- (b) **pdftotext** is then used to convert to the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) **lwarpmk** manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`⁸, `Some-math.html`, and the document's index in `_Index.html`.⁹

2. View the HTML page in a web browser.

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

math

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

3. Force a recompile:

Enter \Rightarrow `lwarpmk again`

Enter \Rightarrow `lwarpmk html`

Enter \Rightarrow `lwarpmk print`

4. Process the HTML index and recompile:¹⁰¹¹

Enter \Rightarrow `lwarpmk htmlindex`

Enter \Rightarrow `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 \Rightarrow `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 9.5.10.

¹¹Also see section 9.5.11 for index options.

6.4 Generating the SVG images

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


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

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


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

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

 **page counter** Incorrect svg images will also occur if the document changes the page counter:

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

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

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

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

6.5 Using MATHJAX for math

math with MATHJAX Math may also be represented using the MATHJAX JAVASCRIPT project.

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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.



MATHJAX requirements

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

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

6.6 Changing the CSS style

For a formal css style, add to the preamble:

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

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

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

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

6.7 Customizing the HTML output

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

See section 8.4 for more information.

6.8 Using latexmk

latexmk is a \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, and a single recompile may be forced with:

```
Enter ⇒ lwarpmk print1
```

and/or

```
Enter ⇒ lwarpmk html1
```

forced single-pass recompile

6.9 Using XeLaTeX or LuaLaTeX

X_gTeX or LuaTeX may be used instead of E_gTeX.

1. Remove the auxiliary files for the project:

```
Enter ⇒ lwarpmk cleanall
```

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

```
Enter ⇒ xelatex tutorial.tex
```

-or-

```
Enter ⇒ lualatex tutorial.tex
```

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

3. To recompile the document:

```
Enter ⇒ lwarpmk print
```

-and-

```
Enter ⇒ lwarpmk html
```

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

```
Enter ⇒ lwarpmk htmlindex
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk printindex
```

```
Enter ⇒ lwarpmk print
```

6.10 Using a glossary

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

Opt `xindyLanguage` To assign the **xindy** language and codepage to be used while processing the index and glossary, use the `xindyLanguage` and `xindyCodepage` options:

```
\usepackage[xindyLanguage=english, xindyCodepage=utf8]{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:

```
Enter ⇒ lwarpmk html1
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print1
```

```
Enter ⇒ lwarpmk print
```

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 Cleaning the images from the `lateximages` directory

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

```
Enter ⇒ lwarpmk cleanimages
```

6.14 Converting PDF images to SVG

HTML cannot display PDF images, so any external PDF graphics images must be converted to `svg` format. `pdftocairo` may be used one image at a time, but `lwarpmk` provides a way to convert images in bulk:

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

6.15 Creating HTML from an incomplete compile

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

```
Enter ⇒ lwarpmk pdftohtml [project]
```

6.16 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
```

6.17 Using the make utility

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

```
lwarpmk pdftohtml [project]
```

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

7 Converting an existing document

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

1. Arrange the document in the following order:
 - (a) Declare the `\documentclass`.
 - (b) Load text fonts.
 - (c) Load **inputenc** or **inputenx**, **fontenc**, and/or **fontspec**.
 - (d) Load **lwarp**.
 - (e) Load remaining packages.
2. Also modify the document:
 - (a) Remove `.pdf` file extensions. Change:


```
\includegraphics{filename.pdf}
```

 to:


```
\includegraphics{filename}
```

 Other image formats may have a file extension.
 - (b) Avoid the `scale` option. Change:


```
\includegraphics[scale=<xx>]
```

 to:


```
\includegraphics[width=<yy>\linewidth]
```
 - (c) Possible changes to tabular environments include `* columns`, **multirow**, **longtable**, **supertabular**, **xtab**, **bigdelim**. See section 9.9.
 - (d) Possible option clashes with **memoir**. See section 9.12.
 - (e) Other changes as per **Special cases and limitations**, section 9.
3. Convert any PDF images to SVG. See section 9.7.
4. Manually compile the print version with **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.

[Need help?](#)

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

8 Additional details

8.1 Shell escape


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

8.2 Font and UTF-8 support

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

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


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

 **pdflatex** `usepackage{lmodern}`


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

`\usepackage{dejavu}`

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

 **xelatex** and **lualatex** Xe_{La}TeX and Lua_{La}TeX 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}`.

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

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

Pkg `fontspec`
`ligatures`

- For \LaTeX or \LuaTeX :
 - **fontspec** and font choices

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}
```

Pkg `lmodern`
 Pkg `fontenc`
 Pkg `inputenc`
 Pkg `inputenx`
 Pkg `newunicodechar`
 File `glyptounicode.tex`

- For **pdf_latex**:
 - (a) `\usepackage{lmodern}`, or other font-related packages
 - (b) `\usepackage[T1]{fontenc}`
 - (c) `\usepackage[utf8]{inputenc}`, or `latin1`, etc. Or use **inputenx**.
 - (d) `\usepackage{newunicodechar}` along with related definitions.
 - (e) To assist with the PDF-HTML conversion:
 - i. `\input glyptounicode.tex`
 - ii. `\input glyptounicode-cmr.tex`% from the pdfx package
 - iii. `\pdfgentounicode=1`
 - (f) Another option to assist with the PDF-HTML conversion:
 - `\usepackage{cmap}`
 - (g) `\usepackage{textcomp}`

3. `\usepackage{lwarp}` (section 8.3) goes after any of the above, followed by:
4. `\usepackage{newtxmath}` or other math-related font packages. Many of these load **amsmath**, which must be loaded after **lwarp**, so they must also be loaded after **lwarp**.
5. `\setmonofont{TeX Gyre Cursor}` or similar may be required if using \LaTeX or \LuaTeX and **fontspec** along with traditional font packages such as **txfonts**, **newtxtext**, etc. This is required to turn off the monospaced font's ligatures with **fontspec** after loading the traditional font packages. Monospaced output ligatures must be turned off to produce the correct HTML characters.
6. ... the rest of the preamble and the main document.

⚠ **fontspec** with
 traditional font packages

8.2.1 Indexes, glossaries, and encoding

lwarp uses the **xindy** program to process indexes. **xelatex** and **lualatex** use **xindy** and **pdf_latex** uses **texindy**.

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

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>latexmk</code>	Boolean for lwarpmk to use latexmk for compiling documents.
<code>lwarpmk</code>	Generate a local copy of <code>lwarpmk.lua</code> .
<code>xindyLanguage</code>	The xindy language option used for index and glossary generation.
<code>xindyCodepage</code>	The xindy codepage option used for index and glossary generation.
<code>xindyStyle</code>	Set a custom filename for xindy .
<code>pdftotextEnc</code>	Set the encoding for pdftotext .

8.3 lwarp package loading and options

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

Pkg	<code>lwarp</code>	Load the lwarp package immediately after the font and UTF-8 setup commands.
Opt	<code>warpprint</code>	Usually controlled by lwarpmk , 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 pdflatex , 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>
Opt	<code>warpHTML</code>	


without any options to create a printed version, while `<project>_html.tex` will create an HTML version.


Opt	<code>mathsvg</code>	For math display, select <code>mathsvg</code> (default), or <code>mathjax</code> . For more information about the math options, see section 9.6.
Opt	<code>mathjax</code>	
Opt	<code>OSWindows</code>	See section 8.6 if using WINDOWS.
Opt	<code>BaseJobname</code>	Not intended for the user. Used internally by lwarp when creating the <code>*_html.tex</code> file used to compile the HTML version. See section 24.
	Default: <code>\jobname</code>	
Opt	<code>HomeHTMLFilename</code>	See section 8.4.
	Default: <code>{}</code>	
Opt	<code>HTMLFilename</code>	See section 8.4.
	Default: <code>{}</code>	
Opt	<code>xindyLanguage</code>	If using an index or glossary, see section 24.
	Default: <code>english</code>	
Opt	<code>xindyCodepage</code>	If using an index or glossary, see section 24.
	Default: <code>utf8</code>	
Opt	<code>latexmk</code>	Tells lwarpmk to use latexmk to recompile the document several times if necessary. Otherwise, lwarpmk attempts to determine for itself whether to recompile. See section 8.4.
	Default: <code>false</code>	
Opt	<code>lwarpmk</code>	If you wish to have lwarp generate a local copy of <code>lwarpmk.lua</code> for archival or local-installation purposes, compile the print version with the <code>lwarpmk</code> option set. See section 24.
Opt	<code>xindyStyle</code>	If you wish to use a custom <code>.xdy</code> file for index generation, see section 24.
	Default: <code>lwarp.xdy</code>	
Opt	<code>pdftotextEnc</code>	Used to specify the encoding used by pdftotext during the PDF-HTML conversion. In most situations, the default is the correct choice.
	Default: <code>UTF-8</code>	

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.
\mathimagename	The svg math image lateximage alt tag.
\packagediagramname	The suffix for a package's lateximage alt tags.
\CSSFilename	The css for the following files.
\HTMLLanguage	The html lang tag.
\HTMLTitle	The HTML title meta tag, overriding \title.
\HTMLAuthor	The HTML author meta tag, overriding \author.
\HTMLDescription	The HTML description meta tag.
\HTMLFirstPageTop	Heading for the home page.
\HTMLPageTop	Heading for the other pages.
\HTMLPageBottom	Footing for all pages.

8.4 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 \Rightarrow `lwarpmk html`

Options for the lwarp package:

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

Opt	<code>HomeHTMLFilename</code> 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 8.4.1 for examples of naming and numbering HTML files.
Opt	<code>HTMLFilename</code> 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 8.4.1 for examples of naming and numbering HTML files.
Opt	<code>latexmk</code> Default: <code>false</code>	latexmk: Controls whether lwarp uses latexmk to compile the document. This setting is written to lwarpmk ’s configuration files.
Opt	<code>mathsvg</code> Default: <code>true</code>	mathsvg: Selects SVG display for math output. (The default.)
Opt	<code>mathjax</code> Default: <code>false</code>	mathjax: Selects MATHJAX for math output.

filename underscores

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

Ctr	<code>tocdepth</code>	tocdepth: Sectioning depth of the table of contents. See section 15 for a list of L ^A T _E X stack depths.
Ctr	<code>SideTOCDepth</code> Default: <code>1</code> <code>sideroc</code>	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: <code>SideTOCDepth=FileDepth</code> or

$\text{SideTOCDepth} = \text{FileDepth} + 1$

inaccessible pages

If $\text{SideTOCDepth} < \text{FileDepth}$, web pages will be inaccessible via the sideroc.

Ctr **FileDepth**
Default: -5

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

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



Bool **CombineHigherDepths**
Default: true

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

The file splits are controlled by the counter FileDepth and the boolean CombineHigherDepths. Setting FileDepth to 0 splits the file at chapters, 1 at sections, etc. CombineHigherDepths controls whether to combine pages at levels higher than the chosen FileDepth, such as in this tutorial where the page which opens the chapter also contains the first section. 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 8.4.1 for examples of naming and numbering HTML files. The user must ensure that filenames are unique after begin sanitized. For example, math in the section name is removed before creating the filename, so the rest of the filename must be sufficiently unique to avoid name collisions.

Unique filename!

Ctr **FootnoteDepth**
Default: 3

FootnoteDepth: Determines where to place pending footnotes. 3 places footnotes before each break down to the \subsubsection level. 1 places

footnotes before each `\section` break. Any pending footnotes are also placed at the bottom of each page before each file break.

`\abstractname`
Default: `Abstract`

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

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

`\CSSFilename`
Default: `lwarp.css`

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

The css styles of the web pages are set by the `\CSSFilename` command. If `\CSSFilename` is not used, a default plain style is used to mimic printed \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.

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

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

 TOC on the homepage! <code>\tableofcontents</code>	<code>\tableofcontents:</code> 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 <code>tocdepth</code> .
--	---

Placed in the document wherever necessary:

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

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

<code>\packagediagramname</code> Default: <code>diagram</code>	<code>\packagediagramname:</code> For many packages, the output is placed inside a <code>lateximage</code> with an HTML <code>alt</code> tag set to the package name followed by <code>\packagediagramname</code> . For example:
---	---

`(-xy- diagram)`

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

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



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

Env `warpHTML`

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



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

`\warpprintonly`

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

`\warpHTMLonly`

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

8.4.1 Example HTML file naming

Examples of ways to name or number HTML files:

Numbered HTML nodes:

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

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

Named HTML sections, no prefix:

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

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

Named HTML sections, with prefix:

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

```
\usepackage[
  HomeHTMLFilename=mywebsite,
```

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

```

        HTMLFilename={mywebsite-}
    ]{lwarp}
    \booltrue{FileSectionNames}

```

8.5 Customizing the CSS

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

```

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

```

The styles provided by **lwarp** include:


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

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

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

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

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

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

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

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

File `project.css`

File `sample_project.css`

```

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

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

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

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

```

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

8.6 Selecting the operating system

Prog `Unix` **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 **lwarp** option `OSWindows`.

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

Opt `OSWindows` the user for later use.

8.7 Selecting actions for print or HTML output


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

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

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

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

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

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

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

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

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

Env `warpall` Anything which is to be done for any output may be surrounded by a `warpall` environment. Doing so is optional.

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

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

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

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

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.

8.8 Commands to be placed into the warpprint environment

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

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

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

Also see section [13: Troubleshooting](#).

8.9 Title page

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

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

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


<code>\maketitle</code>	Use <code>\maketitle</code> just after the <code>\begin{document}</code> , as this will establish the title of the homepage. Optionally, use a <code>titlepage</code> environment instead.
Env <code>titlepage</code>	The <code>titlepage</code> environment may be used to hold a custom title page. The <code>titlepage</code> will be set in a <code><div></code> class <code>titlepage</code> , and <code>\printtitle</code> , etc. may be used inside this environment.
Env <code>titlingpage</code>	Another form of custom title page, where <code>\maketitle</code> is allowed, and additional information may be included as well.
<code>\title</code>	<code>{\title}</code>



Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use a `\subtitle` command instead (section [62.8](#)). The title will appear in the

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

`\author` `{\langle author \rangle}`

 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` `{\langle date \rangle}`

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

`\thanks` `{\langle text \rangle}`

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

8.10 HTML page meta descriptions

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

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

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

disabling To disable the generation of HTML description meta tags, use:
`\HTMLDescription{}`

8.11 HTML page meta title

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

8.12 HTML page meta author

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

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

8.13 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 xindyStyle 3. In the document source use the `xindyStyle` option for **lwarp**:

```
\usepackage[
... other options ...
xindyStyle=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`.

9 Special cases and limitations

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

[Need help?](#)

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

9.1 Things to avoid

In the document, avoid the following:

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

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

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

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

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

9.2 Formatting

9.2.1 Text formatting

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

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

9.2.2 Horizontal space

`\hspace` `\hspace` is converted to an inline HTML span of the given width, except that 0 width is

ignored, a width of `.16667em` is converted to an HTML thin breakable space (U+2009), and a `\fill` is converted to a `\qqquad`.

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

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

9.2.3 Text alignment

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

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


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

9.2.4 Accents

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

9.2.5 textcomp package

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

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

9.2.6 Superscripts and other non-math uses of math mode

Use `x` instead of `x`

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

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

9.2.8 Filenames and URLs in lists or footnotes

filename underscore Escape underscores in the filenames:


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

9.2.9 relsize package

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

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

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

9.3 Boxes and minipages

9.3.1 Marginpars



\marginpar [*left*] {*right*} `\marginpar` may contain 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.

9.3.2 Save Boxes

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

9.3.3 Minipages

-  **inline** A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made inline-block.
- placement** Minipages and parboxes will be placed side-by-side in HTML unless you place a `\newline` between them.
- side-by-side** Side-by-side minipages may be separated by `\quad`, `\qqquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a center environment, the result is similar in print and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.
- in a span** There is limited support for minipages inside an HTML ``. An HTML `<div>` cannot appear inside a ``. While in a ``, minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an “inline” format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.
- size** When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.
- no-width minipages** A minipage of width exactly `\linewidth` is automatically given no HTML width.
- full-width minipages** A new macro `\minipagefullwidth` requests that the next minipage be generated without an HTML width attribute, allowing it to be the full width of the display rather than the fixed width given.
-  **text alignment** Nested minipages adopt their parent’s text alignment in HTML, whereas in regular \TeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

9.3.4 Side-by-side minipages

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

9.3.5 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 370 for `xcolor`'s `\colorbox` and `\fcolorbox`, and `lwarp`'s additional `\colorboxBlock` and `\fcolorboxBlock`.

⚠ Misplaced alignment
 tab character &

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

To frame equations: See section 174 for the `fancybox` package.

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

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

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

9.3.6 fancybox package

Pkg fancybox
framed equation example

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

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

framing alternatives

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

framed table example

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

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

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

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

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

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

9.3.7 mdframed package

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

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

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

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

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

ignored options `userdefinedwidth` and `align` are currently ignored.

9.4 Cross-references


labels 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 `\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.

 **empty link**

9.4.1 Page references

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

Ex:

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

`\pagerefPageFor` may be redefined to “page for”, empty, etc. See page 431.

9.4.2 cleveref and varioref packages

Pkg cleveref `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”.

Pkg varioref

 **cleveref page numbers**

Ex:

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

See `\cpagerefFor` at page 513 to redefine the message which is printed for page number references.

9.4.3 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.

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

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

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

9.4.4 Footnotes and page notes

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

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

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

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

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

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

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

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

⚠ **sectioning or displaymath**

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

and likewise for equations or display math.

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



pfnote numbers

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

9.5 Front and back matter

9.5.1 Custom classes with multiple authors and affiliations

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

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

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

9.5.2 Starred chapters and sections

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

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

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

\ForceHTMLPage

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

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

⚠ inaccessible HTML page

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

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

9.5.3 abstract package

Pkg abstract

⚠ missing TOC

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

9.5.4 titling and authblk

Pkg titling

Pkg authblk

package support

⚠ load order

`\published` and `\subtitle`

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

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

9.5.5 tocloft package

Opt tocloft titles

Pkg tocloft

Pkg tocloft

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

⚠ tocloft & other packages

9.5.6 appendix package

Pkg appendix

⚠ incorrect TOC link

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

9.5.7 pagenote package

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

9.5.8 endnotes package

Pkg endnotes
table of contents To place the endnotes in the TOC, use:

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

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

```
\ForceHTMLPage
\theendnotes
```

9.5.9 BibTeX

\etalchar Displays a superscript “+” to indicate “and others”.

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

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

9.5.10 glossaries package

Pkg glossaries **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	xindyLanguage	The lwarp package option <code>xindyLanguage</code> sets the language used by xindy . This is passed to xindy using its <code>-L</code> option, and is used for both index and glossary generation.
	Default: <code>english</code>	
Opt	xindyCodepage	The option <code>xindycodepage</code> sets the codepage used by xindy . This is passed to xindy using its <code>-C</code> option, and is used for both index and glossary generation.
	Default: <code>utf8</code>	
Opt lwarpmk	printglossary	lwarpmk has the commands <code>lwarpmk printglossary</code> and <code>lwarpmk htmlglossary</code> to process the glossaries created by glossaries using xindy .
Opt lwarpmk	htmlglossary	

9.5.11 Index and the tocbibind package

Opt	xindyLanguage	The lwarp package option <code>xindyLanguage</code> sets the language used by xindy . This is passed to xindy using its <code>-L</code> option, and is used for both index and glossary generation.
	Default: <code>english</code>	
Opt	xindyCodepage	The option <code>xindycodepage</code> sets the codepage used by xindy . This is passed to xindy using its <code>-C</code> option, and is used for both index and glossary generation.
	Default: <code>utf8</code>	
Pkg	tocloft	If using tocloft with tocbibind , anonchapp , fncychap , or other packages which change chapter title formatting, load tocloft with its <code>titles</code> option, which tells tocloft to use standard \LaTeX commands to create the titles, allowing other packages to work with it.
 tocloft & other packages		
Pkg	makeidx	An index may be placed inline with other HTML text, or on its own HTML page:
placement and toc options		

Inline, with a manual TOC entry:

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

```

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

```

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

```

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

```

Inline, with an automatic TOC entry:

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

```

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

```

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

```

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

```

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

[numbered index section](#)

See section [72](#) for **lwarp**'s core index and glossary code, and section [346](#) for **tocbibind**.

9.6 Math

9.6.1 Rendering tradeoffs

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

[SVG files](#) Rendering math as images creates a new svg file for each expression, except that an

MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to svg only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change.

- SVG inline** The svg images are currently stored separately, but they could be encoded in-line directly into the HTML document. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded.
- PNG files** Others \LaTeX -to-HTML converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but svg files are the preferred approach for scalable graphics.
- MathML** Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than svg drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 10 regarding EPUB output with MATHJAX.

9.6.2 SVG option

- SVG math option** For svg math, math is rendered as usual by \LaTeX into the initial PDF file using the current font¹⁴, then is captured from the PDF and converted to svg graphics via a number of utility programs. The svg format is a scalable-vector web format, so math may be typeset by \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML alt attribute carries the \LaTeX code which generated the math, allowing copy/paste of the \LaTeX math expression into other documents.

- SVG image font size** For the `lateximage` environment, the size of the math and text used in the svg image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, which defaults to:

```
\renewcommand{\LateximageFontSizeName}{normalsize}
```

For inline svg math, font size is instead controlled by `\LateximageFontScale`, which defaults to:

```
\newcommand*{\LateximageFontScale}{.75}
```

- SVG math copy/paste** For svg math, text copy/paste from the HTML `<alt>` tags lists the equation number or tag for single equations, along with the \LaTeX code for the math expression. For \mathcal{AMS} environments with multiple numbers in the same environment, only the first and

¹⁴See section 372 regarding fonts and fractions.

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.

9.6.3 MATHJAX option

MATHJAX math option

Prog MathJax

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

When MATHJAX is enabled, math is rendered twice:

1. As regular \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.

9.6.4 Customizing MATHJAX

MATHJAX does not have preexisting support every possible math function. Additional MATHJAX function definitions may be defined. These will be declared at the start of each HTML page, and thus will have a global effect.

Examples:

```
\CustomizeMathJax{
  \newcommand{\expval}[1]{\langle#1\rangle}
  \newcommand{\abs}[1]{\lvert#1\rvert}
}
\CustomizeMathJax{\newcommand{\arsinh}{\text{arsinh}}}
\CustomizeMathJax{\newcommand{\arcosh}{\text{arcosh}}}
\CustomizeMathJax{\newcommand{\NN}{\mathbb{N}}}
```

9.6.5 MATHJAX limitations

MATHJAX limitations

Prog MathJax

chapter numbers

Limitations when using MATHJAX include:

- In document classes which have chapters, `\tagged` equations have the chapter

number prepended in HTML output, unlike \LaTeX . $\backslash tag*$ 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

⚠ siunitx inside an equation

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

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

Also see section 9.6.12.

tabbing

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

⚠ other macros and packages

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

9.6.6 Catcode changes

preamble macros with math

The math shift character `$` is not set for HTML output until after the preamble. Macros defined in the preamble which contain `$` must be enclosed between `\StartDefiningMath` and `\StopDefiningMath` to temporarily change to the HTML meaning of `$`:

```
\StartDefiningMath
\newcommand{...}
\StopDefiningMath
```

As an alternative, use `\(` and `\)` instead of `$`, in which case `\StartDefiningMath` and `\StopDefiningMath` are not necessary.

If a package defines macros using `$`, it may be necessary to use `\StartDefiningMath` and `\StopDefiningMath` before and after loading the package.

9.6.7 Dynamic math


inline dynamic math An inline math expression is usually converted to a reusable hashed svg math image, or a MathJax expression. The hash or expression depends on the contents of the math expression. In most cases this math expression is static, such as $x+1$, so the image can be reused for multiples instances of the same expression. In some cases, the math expression includes a counter or other object which may change between uses. The macro `\StartDynamicMath` may be used before a dynamic math expression, and `\StopDynamicMath` after. Doing so tells **lwarp** to use an unhashed svg math image, even if MathJax is in use. See section 41.

9.6.8 Display math

`\displaymathnormal` By default, or when selecting `\displaymathnormal`, math display environments print their contents in MATHJAX, and render their contents in svg math as well as use their contents in the alt tag of HTML output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated Tikz pictures, compilation will fail.

`\displaymathother` When selecting `\displaymathother`, it is assumed that the contents are more complicated than “pure” math. An example is an elaborate Tikz picture, which will not render in MATHJAX and will not make sense as an HTML alt tag. In this mode, MATHJAX is turned off, math display environments become svg images, even for MATHJAX, and the HTML alt tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as Tikz pictures are more likely to compile successfully.

9.6.9 chemformula package

 **chemformula with MATHJAX** **chemformula** works best without MATHJAX. If MATHJAX is used, `\displaymathother` must be used before `array`, and then `\displaymathnormal` may be used after. (The **chemformula** package adapts to `array`, but does not know about MATHJAX, and MATHJAX does not know about **chemformula**.)

While using MATHJAX, `\displaymathother` may also be used for other forms of display and inline math which contain **chemformula** expressions.

9.6.10 mhchem package

See section 251.

9.6.11 ntheorem package

Pkg `ntheorem`

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

⚠ Font control

⚠ Equation numbering

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

9.6.12 siunitx package

Pkg `siunitx`

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

`fractions`

⚠ math mode required

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

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

⚠ tabular

Tabular `S` columns are rendered as simple `c` columns, and tabular `s` columns are not supported. These may be replaced by `c` columns with each cell contained in `\num` or `\si`.

9.6.13 units and nicefrac packages

Pkg `units`

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

Pkg `nicefrac`

9.6.14 newtxmath package

Pkg `newtxmath`

The proper load order is:

⚠ loading sequence

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

9.7 Graphics

Pkg `graphics` 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:**

⚠ `.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, or 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
```

For a large number of images, use **lwarpmk**:

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

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** 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. Do not use the `scale` option, since it is not well supported by HTML browsers.

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

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

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

⚠ **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The css3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \TeX , so expect some ugly results for scaling and rotating.

9.7.1 tikz package

Pkg `tikz` If using display math with `tikzpicture` or `\tikz`, along with matrices with the `&` character, the document must be modified as follows:

⚠ **displaymath and matrices**

```
\usepackage{tikz}
\tikzset{every picture/.style={ampersand replacement=\&}}
```

and each instance of `&` in the **tikz** expression must be replaced with `\&`.

9.7.2 grffile package

Pkg `grffile` **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**

9.7.3 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.

9.7.4 xcolor package

Pkg `xcolor` `\colorboxBlock` and `\fcolorboxBlock` are provided for increased HTML compatibility, and they are identical to `\colorbox` and `\fcolorbox` in print mode. In HTML mode they place their contents into a `<div>` instead of a ``. These `<div>`s are set to display: `inline-block` so adjacent `\colorboxBlock`s appear side-by-side in HTML, although text is placed before or after each.

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

`background: none` `\fcolorbox` and `\fcolorboxBlock` allow a background color of `none`, in which case

only the frame is drawn, which can be useful for HTML.

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

colored tables `\rowcolors` is supported, except that the optional argument is ignored so far.

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


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

9.7.5 epstopdf package

Pkg `epstopdf` 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.

9.7.6 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**

9.8 Tabbing

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

math in tabbing If math is used inside tabbing, place tabbing inside a `lateximage` environment, which will render the entire environment as a single SVG image.

9.9 Tabular

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, * column specifiers, `siunitx` S columns, or the packages **multirow**, **longtable**, **supertabular**, or **xtab**.

Defining environments:

⚠ misplaced alignment
alignment tab character &

⚠ 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>
\StopDefiningTabulars
```

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
}
\StopDefiningTabulars
```

Cell contents:

⚠ paragraphs

⚠ \multirow

- 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 & ...
```

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.

⚠ `\multicolumn &`
`\multirow`

⚠ skipped cells

⚠ empty cells

`vposn`

⚠ macro in a table
custom macros

- See section 260.2 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`

⚠ `\newcolumnntype`

- * 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.
- `\newcolumnntype` is ignored; unknown column types are set to 1.

Rules:

vertical rules

width and trim

full-width rules

- Doubled `\hlines`, `\midrules`, and vertical rules are supported.
- 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

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

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

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

⚠ `\warpprintonly`
misplaced `\noalign`

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

```
This & That \endhead
```

to

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

and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.

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 \\}
```

9.9.1 longtable package

Pkg `longtable`

⚠

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

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

instead of

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

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

Keep the \endfirsthead row, which is still 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.

9.9.2 supertabular and xtab packages

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

```
Pkg xtab \StartDefiningTabulars
\tablefirsthead
...
\StopDefiningTabulars
```

⚠ misplaced alignment
alignment tab character &

See section 9.9.

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

9.9.3 bigdelim package

Pkg bigdelim \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:

⚠ use \mrowcell

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

```
<>    a    b
left {  c    d
      {  e    f
<>    g    h
```

9.10 Floats

9.10.1 Float contents alignment

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

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

9.10.2 float, trivfloat, and/or algorithmicx together

`Pkg float` If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section 354.1.

`Pkg trivfloat`

`Pkg algorithmicx`

 **package conflicts**

9.10.3 caption and subcaption packages

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

`Pkg subcaption`

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

 **options**


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

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

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

9.10.4 subfig package

`Pkg subfig`

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

horizontal spacing In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph

tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

9.10.5 floatrow package

Pkg floatrow

⚠ misplaced alignment
alignment tab character &

⚠ subfig package

Use `\StartDefiningTabulars` and `\StopDefiningTabulars` before and after defining macros using `\ttabbox` with a tabular inside. See section 9.9.

When combined with the **subfig** package, while inside a `subfloatrow` `\ffigbox` and `\ttabbox` must have the caption in the first of the two of the mandatory arguments.

⚠ `\FBwidth`, `\FBheight`

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

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

9.10.6 keyfloat package

Pkg keyfloat

⚠ keywrap

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

9.11 Koma-Script

Cls komascript


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

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

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

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

9.12 Memoir

 **options clash** While emulating **memoir**, **lwarp** pre-loads a number of packages (section 382.1). This can cause an options clash when the user's document later loads the same packages with options. To fix this problem, specify the options before loading **lwarp**:

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

`\verbfootnote` is not supported.

`\newfootnoteseries`, etc. are not supported.

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

Poem numbering is not supported.

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

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

9.13 Miscellaneous

9.13.1 verse and memoir

Pkg **verse** When using **verse** or **memoir**, always place a `\\` after each line.

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

```


\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```

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

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

9.13.2 newclude package

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

 **loading**

```

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

```

9.13.3 babel package

Pkg **babel**

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

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

punctuation spaces Also when French is used, **lwarp** creates fixed-width space around punctuation by patching `\FBcolonspace`, `\FBthinspace`, `\FBguillspace`, `\FBmedkern`, `\FBthickkern`,

customized spacing

`\FBtextellipsis`, and the tilde. If the user's document also changes these parameters, the user's changes should be placed inside a `warpprint` environment so that the user's changes do not affect the HTML output.

9.13.4 polyglossia package

Pkg `polyglossia` **lwarp** uses **cleveref**, which has some limitations when using **polyglossia**, possibly resulting in the error

```
! Undefined control sequence. ... \@begindocumenthook
```

To test compatibility, add

```
\usepackage{cleveref}
```

near the end of the preamble (as the last package to be loaded), and try to compile the print version. It may be necessary to set

```
\setdefaultlanguage{english}
```

or some other language supported by **cleveref**, then select other languages using `\setotherlanguages`.

Once the print version works with **cleveref** and **polyglossia**, the HTML version should work as well using **lwarp**.

9.13.5 todonotes and luatodonotes packages

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

9.13.6 fixme

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

external layouts

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

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

9.13.7 xparse

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

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

10 EPUB conversion

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

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

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

<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

⚠ encoding

Check the box Add linked files in breadth first order. Set the document encoding as `utf-8`, which is what **lwarp** generates for HTML, even if the original printed document uses some other encoding.

⚠ section breaks

The EPUB-conversion program must also know where the section breaks are located. For a list of **lwarp**'s section headings, see table 7. For example, an `article` class document would break at `\section`, which is mapped to HTML heading level `<h4>`, whereas a `book` class document would break at `\chapter`, which is HTML heading level `<h3>`. For **Calibre**, this option is found in

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

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

Match HTML tags with tag name:

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

`//h:h4` — or — `//h:h3`

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

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

MATHJAX support

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

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

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

11 Word-processor conversion

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

11.1 Activating word-processor conversion

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

Bool	FormatWP
Default: false	<hr/> <i>FormatWP</i> <hr/> 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. <hr/>

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.

11.2 Additional modifications

WPMarkFloats

Adds

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

or

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

Bool WPMarkFloats
Default: false

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

WPMarkMinipages

Adds

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

Bool WPMarkMinipages
Default: false

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

WPMarkTOC

While formatting for word processors, adds

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

Bool WPMarkTOC
Default: true

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

If set false, the actual TOC is printed instead.

WPMarkLOFT

While formatting for word processors, adds

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

Bool WPMarkLOFT
Default: false

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

If set false, the actual lists are printed instead.

WPMarkMath

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

Bool WPMarkMath
Default: false
Prog TeXMaths
siunitx

When using the **siunitx** package, enter

```
\usepackage{siunitx}
```

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

WPTitleHeading

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

Bool WPTitleHeading
Default: false
section headings

See table 6 on page 148.

11.3 Recommendations

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

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

Table 6: Section HTML headings for word-processor conversion

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 page 160.

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

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

siunitx For **siunitx**, remember to adjust the preamble as mentioned above.

LO view border options LIBREOFFICE has options in the **View** menu to turn on/off the display of thin borders around table cells and text objects.

11.4 Limitations

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

Cross references are not actual LIBREOFFICE linked cross references.

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

Equation numbering is not automatic, but the equation numbers in svg math will match the \LaTeX and HTML output. SVG math is recommended when using the \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.

12 Modifying lwarp

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

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


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

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

image of T_EX output An environment may be converted to a `lateximage` then displayed with an image of the resulting T_EX output. See section 84 for an example of the `picture` environment.

CSS classes To create a custom HTML block or inline CSS class, see section 49.8.

print/HTML macros To create print and HTML versions of the same macro or environment, see section 33.

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

index recreation To recreate the index for the **lwarp** documentation:

```
makeindex -s gglo.ist -o lwarp.gls lwarp.glo
makeindex -s gind.ist lwarp.idx
```

12.1 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 T_EX 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 T_EX 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 273 ([ntheorem](#)) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable without changes, there is no need to create a `lwarp-` version.

12.1.1 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 25.1. This causes **lwarp** to stop with an error if `packagename` is loaded before **lwarp**.

12.2 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.

12.3 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.

When changing the source to **lwarpmk** or a CSS file in `lwarp.dtx`:

1. Change the source in `lwarp.dtx`.
2. `pdflatex lwarp.ins`
3. `pdflatex lwarp.dtx`
4. If modifying **lwarpmk** the new version should now be active.
5. If modifying CSS files:
 - (a) For the document, `lwarpmk print` to update the CSS files in the project.
 - (b) Reload the HTML document to see the effect of the new CSS files.

Sometimes it is worth checking the `<project>_html.pdf` file, which is the PDF containing HTML tags. Also, `<project>_html.html` has the text conversion of these tags, before the file is split into individual HTML files.

It is also worth checking the browser's tools for verifying the correctness of HTML and CSS code.

12.4 Modifying lwarpmk

Prog <code>lwarpmk</code> File <code>lwarpmk.lua</code>	In most installations, <code>lwarpmk.lua</code> is an executable file located somewhere the operating system knows about, and it is called by typing “ <code>lwarpmk</code> ” 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`”.

13 Troubleshooting

13.1 Using the lwarp.sty package

Also see:

Section 8.8: [Commands to be placed into the warpprint environment](#)

Section 9: [Special cases and limitations](#)

Text is not converting:

- Font-related UTF-8 information must be embedded in the PDF file. See section 8.2 regarding vector fonts.

Undefined HTML settings:

- See the warning regarding the placement of the HTML settings at section 8.4.

Tabular problems: See section 9.9.

Obscure error messages:

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

`\end{warpHTML}`, `\end{warpprint}`, `\end{warpall}`: Each of these must be without any other characters on the same line.

Options clash: If using `memoir`, see section 9.12.

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

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

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

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

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

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

If you find something using `\ensuremath`, have it temporarily set:

```
\LetLtxMacro\@ensuredmath\LWR@origensuredmath
```

inside a group first.

Also, custom macros which appear inside a section, figure, or table name should be made robust since they appear inside the `.toc`, `.lof`, or `.lot` files. Use `\newrobustcmd` or `\robustify` from **etoolbox**, **xparse**, etc.

If using BibTeX, see section 9.5.9.

⚠ custom macros in
section names

⚠ BibTeX

⚠ polyglossia

“! Undefined control sequence. ... \@begindocumenthook”: See section 9.13.4 if using **polyglossia**.

⚠ custom macros for
environments

“\begin{equation} ended by \end{document}”: Do not use custom macros such as `\beq` and `\eeq` to replace

```
\begin{equation}
```

```
...
```

```
\end{equation}
```

⚠ display math

Complicated objects inside display math: Some objects, such as `Tikz`, may not compile in **lwarp**’s normal display math emulation. Insert

```
\displaymathother
```

before the display math environment, and then

```
\displaymathnormal
```

when displaying “normal” math. See section 9.6.8.

⚠ MathJax

Incorrect MATHJAX: Some objects do not convert to MATHJAX. Use `\displaymathother` before these objects, then `\displaymathnormal` to return to “normal” display math. See section 9.6.8.

Missing sections: See section 8.4 regarding the `FileDepth` and `SideTOCDepth` counters, and the use of `\tableofcontents` in the home page.

Misnumbered footnotes from section headings: See section 9.4.4.

Missing HTML files:


- See the warning regarding changes to the HTML settings at section 8.4.
- Ensure that the filenames are unique after math and short words are removed. See `FileSectionNames` at section 8.4.

Missing / incorrect cross-references:

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

labels

⚠ underscores

 `\nameref`
empty link

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

 `cleveref` page numbers

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

Ex:

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

in HTML becomes:

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

See `\cpagerefFor` at page 513 to redefine the message which is printed for page number references.

BibTeX errors with `\etalchar`: See section 9.5.9.

Malformed URLs: Do not use the % character between arguments of `\hyperref`, etc., as this character is among those which is neutralized for inclusion in HTML URLs.

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

Use `XYLaTeX` or `LuaYLaTeX`.

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:

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

SVG images:

 adding/removing

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

Before attempting to create the svg image files, `lwarpmk` verifies that the HTML version of the document exists and has correct internal image references.¹⁵ If it

¹⁵This becomes important when dealing with a document containing thousands of images.

is necessary to recompile the document's HTML version, **lwarpmk** will inform so with an error message.

⚠ HTML instead of images

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

⚠ page counter

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

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

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

⚠ Lots of files!

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

Plain-looking document:

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

Broken fragments of HTML:

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

Changes do not seem to be taking effect:

- Be sure to `lwarpmk clean`, recompile, then start by reloading the home page. You may have been looking at an older version of the document. If you changed a section name, you may have been looking at the file for the old name.
- See the warning regarding changes to the HTML settings at section 8.4.
- Verify that the proper CSS is actually being used.
- The browser may compensate for some subtle changes, such as automatically generating ligatures, reflowing text, etc.

Un-matched conditional compiles:

- Verify the proper `begin/end` of `warpprint`, `warpHTML`, and `warpall` environments.

13.1.1 Debug tracing output

`\tracinglwarp` When `\tracinglwarp` is used, **lwarp** will add extra tracing messages to the `.log` file. The last several messages may help track down errors.

Place `\tracinglwarp` just after `\usepackage{lwarp}` to activate tracing.

13.2 Compiling the `lwarp.dtx` file

lwarp_tutorial.tex: Copy or link `lwarp_tutorial.txt` from the TDS doc directory to the source directory, or wherever you wish to compile the documentation. This file is included verbatim in the documentation, but is in the doc directory so that it may be found by **texdoc** and copied by the user.

Illogical error messages caused by an out-of-sync `lwarp.sty` file:

1. Delete the `lwarp.sty` file.
2. Enter `pdflatex lwarp.ins` to generate a new `lwarp.sty` file.
3. Enter `pdflatex lwarp.dtx` to recompile the `lwarp.pdf` documentation.

Un-nested environments:

Be sure to properly nest:

- `\begin{macrocode}` and `\end{macrocode}`
- `\begin{macro}` and `\end{macro}`
- `\begin{environment}` and `\end{environment}`

File 1 **lwarp.sty**

14 Implementation

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

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

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

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

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

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

Table 7: Section depths and HTML headings

Section	\LaTeX depth	HTML headings *
title of the entire website		<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 page 148.

15 Section depths and HTML headings

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

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

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

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

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

16 Source Code

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


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

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

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

objects Black-colored tags in the left 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:

17 Detecting the TeX Engine — pdf_latex, lua_latex, xe_latex

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

Detects Xe_lTeX and Lua_lTeX:

```
1 \RequirePackage{iftex}
2 \newif\ifxetexorluatex
3 \ifXeTeX
4   \xetexorluatextrue
5 \else
6   \ifLuaTeX
7     \xetexorluatextrue
8   \else
9     \xetexorluatexfalse
10  \fi
11 \fi
12
13 \ifLuaTeX
14 \RequirePackage{luatex85}% until the geometry package is updated
15 \fi
```

18 MD5 hashing

The MD5 hash is used for lateximage filenames for svg math.

```
16 \newcommand{\LWR@mdfive}[1]{%
17 \PackageError{lwarp}
18 {No MD5 macro was found.}
19 {Lwarp must find the macros pdfmdfivesum or mdfivesum.}
20 }
21
22 \ifPDFTeX
23 \let\LWR@mdfive\pdfmdfivesum
24 \fi
25
26 \ifLuaTeX
27 \RequirePackage{pdftexcmds}
28 \let\LWR@mdfive\pdf@mdfivesum
29 \fi
30
31 \ifXeTeX
32 \@ifundefined{pdf@mdfivesum}{%
33   {\let\LWR@mdfive\pdfmdfivesum}
```

```

34 \@ifundefined{mdfivesum}{}
35     {\let\LWR@mdfive\mdfivesum}
36 \fi

```

19 pdfLaTeX T1 and UTF8 encoding

When using pdf~~La~~TeX, **lwarp** requires T1 encoding, and recommends UTF8 encoding.

If some other input encoding is already defined, **lwarp** will try to use it instead, and hope for the best.

X~~La~~TeX and Lua~~La~~TeX are both UTF8 by nature.

```

37 \ifPDFTeX
38 \RequirePackage[T1]{fontenc}
39
40 \@ifpackageloaded{inputenc}{}{
41     \@ifpackageloaded{inputenx}{}{
42         \RequirePackage[utf8]{inputenc}
43     }
44 }
45 \fi

```

20 Unicode input characters

for HTML & PRINT: If using **pdflatex**, convert a minimal set of Unicode characters. Additional characters may be defined by the user, as needed.

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

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

```

46
47 \RequirePackage{newunicodechar}
48
49 \newunicodechar{×}{\texttimes}
50
51 \ifPDFTeX
52 \newunicodechar{ff}{ff}% the first arguments are ligatures
53 \newunicodechar{fi}{fi}
54 \newunicodechar{fl}{fl}
55 \newunicodechar{ffi}{ffi}

```

```

56 \newunicodechar{ffl}{ffl}
57 \newunicodechar{--}{---}
58 \newunicodechar{-}{--}

```

In PDF_T_EX, preserve upright quotes in verbatim text:

```

59 \RequirePackage{upquote}
60 \else
61 \fi

```

macrocode

21 Miscellaneous tools

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

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

```

62 \newcommand*\LWR@providelength[1]{%
63   \ifdeflength{#1}{\newlength{#1}}%
64 }

```

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

`\LWR@convertto` `{\dest unit}` `{\length}`

```

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

```

22 Early package requirements

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

`Pkg xpatch` Patches macros with optional arguments.

```

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

```

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

```

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

```

Pkg letltxmacro Used to redefine \textbf and friends.

```
69 \RequirePackage{letltxmacro}
```

23 Operating-System portability

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

23.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

```
70 \newbool{usingOSWindows}
71 \boolfalse{usingOSWindows}
```

23.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

23.3 MS-WINDOWS

For MS-WINDOWS:

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

```

73 \newcommand*{\LWR@setOSWindows}
74 {
75 \booltrue{usingOSWindows}
76 \renewcommand*{\OSPathSymbol}{\@backslashchar}
77 }

```

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

```

78 \ifwindows
79 \LWR@setOSWindows
80 \fi

```

24 Package options

Pkg **kvoptions** Allows key/value package options.

```

81 \RequirePackage{kvoptions}
82 \SetupKeyvalOptions{family=LWR,prefix=LWR@}

```

Bool **warpingprint**

Bool **warpingHTML**

Bool **mathjax**

Bool **LWR@origmathjax**

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

LWR@origmathjax remembers the original setting to be restored by `\displaymathnormal`.

```

83 \newbool{warpingprint}
84 \newbool{warpingHTML}
85 \newbool{mathjax}
86 \newbool{LWR@origmathjax}

```

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

```

87 \booltrue{warpingprint}%
88 \boolfalse{warpingHTML}%
89 \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.

```

90 \DeclareVoidOption{warpprint}{%
91 \PackageInfo{lwarp}{Using option 'warpprint'}
92 \booltrue{warpingprint}%
93 \boolfalse{warpingHTML}%

```

94 }

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.

```
95 \DeclareVoidOption{warpHTML}{%
96 \PackageInfo{lwarp}{Using option 'warpHTML'}%
97 \booltrue{warpingHTML}%
98 \boolfalse{warpingprint}%
99 }
```

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.

```
100 \DeclareVoidOption{mathsvg}{%
101 \PackageInfo{lwarp}{Using option 'mathsvg'}
102 \boolfalse{mathjax}%
103 \boolfalse{LWR@origmathjax}%
104 }
```

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

```
105 \DeclareVoidOption{mathjax}{%
106 \PackageInfo{lwarp}{Using option 'mathjax'}
107 \booltrue{mathjax}%
108 \booltrue{LWR@origmathjax}%
109 }
```

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.

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

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

```
111 \DeclareStringOption[english]{xindyLanguage}
```

Opt **xindyCodepage** Sets the **xindy** codepage to be assigned in **lwarpmk**'s configuration files. This is then

used by **lwarpmk** while processing the index and glossary.

```
112 \DeclareStringOption[utf8]{xindyCodepage}
```

Opt xindyStyle Selects a custom .xdy file. The default is lwarp.xdy. A customized file should be based on lwarp.xdy, and must retain the line

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

```
113 \DeclareStringOption[lwarp.xdy]{xindyStyle}
```

Opt pdftotextEnc The option pdftotextEnc sets the encoding used by **pdftotext**. This is passed to **pdftotext** using its -enc option, and is used when converting \LaTeX PDF output with HTML tags into a plain-text file with HTML tags.

Default: UTF-8

```
114 \DeclareStringOption[UTF-8]{pdftotextEnc}
```

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

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

```
115 \newbool{LWR@creatinglwarpmk}
116 \boolfalse{LWR@creatinglwarpmk}
117
118 \DeclareVoidOption{lwarpmk}{
119 \PackageInfo{lwarp}{Using option 'lwarpmk'}
120 \booltrue{LWR@creatinglwarpmk}
121 }
```

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.

```
122 \DeclareVoidOption{OSWindows}{
123 \PackageInfo{lwarp}{Using option 'OSWindows'}
124 \LWR@setOSWindows
125 }
```

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

```
126 \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>

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

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

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

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

```
129 \ProcessKeyvalOptions*\relax
```

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

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

Defaults unless already over-ridden by the user:

```
131 \ifcsemtty{\LWR@HomeHTMLFilename}{
132 \newcommand*{\HomeHTMLFilename}{\BaseJobname}
133 }{
134 \csedef{\HomeHTMLFilename}{\LWR@HomeHTMLFilename}
135 }
136
137 \csedef{\HTMLFilename}{\LWR@HTMLFilename}
```

24.1 Conditional compilation

\warpprintonly {<contents>}

Only process the contents if producing printed output.

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

\warpHTMLonly {<contents>}

Only process the contents if producing HTML output.

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

Pkg comment Provides conditional code blocks.

```
140 \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.

```

141 \ifbool{warpingHTML}{
142 \def\DefaultCutFileName{\def\CommentCutFile{comment_html.cut}}
143 }{}
144
145 \ifbool{warpingprint}{
146 \def\DefaultCutFileName{\def\CommentCutFile{comment_print.cut}}
147 }{}

148 \excludecomment{testing}

```

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

```

149 \includecomment{warpall}

```

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

Env **warpHTML**

For HTML output:

```

150 \ifbool{warpingHTML}{%
151 \includecomment{warpHTML}
152 }
153 {\excludecomment{warpHTML}}%

154 \ifbool{warpingprint}
155 {\includecomment{warpprint}}
156 {\excludecomment{warpprint}}

```

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

```

157 \ifbool{LWR@creatinglwarpmk}
158 {\includecomment{LWR@createlwarpmk}}
159 {\excludecomment{LWR@createlwarpmk}}

```

25 Package load order

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

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

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

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

Most packages should be loaded after **lwarp**. This is 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.

25.1 Tests of package load order

`\LWR@loadafter` $\{\langle packagename \rangle\}$ Error if this package was loaded before **lwarp**.

```
160 \newcommand*\LWR@loadafter}[1]{%
161 \@ifpackageloaded{#1}
162 {
163 \PackageError{lwarp}
164 {Package #1, or one which uses #1, must be loaded after lwarp}
165 {Move \detokenize{\usepackage}{#1} after \detokenize{\usepackage}{lwarp}.
166 Package #1 may also be loaded by something else, which must also be moved
167 after lwarp.}
168 }
169 {}
170 }
```

`\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.

```
171 \ifclassloaded{memoir}
172 {\newcommand*\LWR@notmemoirloadafter}[1]{}
173 {\LetLtxMacro\LWR@notmemoirloadafter\LWR@loadafter}}
```

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

```
174 \newcommand*\LWR@loadbefore}[1]{%
175 \@ifpackageloaded{#1}
176 {}
177 {
178 \PackageError{lwarp}
179 {Package #1 must be loaded before lwarp}
180 {Move \detokenize{\usepackage}{#1} before \detokenize{\usepackage}{lwarp}.}
181 }
182 }
```

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

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

```
183 \newcommand*\LWR@loadnever[2]{%
184 \PackageError{lwarp}
185 {Package #1 is not supported by lwarp's HTML conversion.
186 Package(s) #2 may be useful instead}
187 {Package #1 might conflict with lwarp in some way,
188 or is superceded by another package.
189 For a possible alternative, see package(s) #2.}
190 }
```

`\LWR@earlyloadnever` `{\badpackagename}{\replacementpkgname}`

The first packages is not supported, so tell the user to use the second instead. This version checks immediately for packages which may have been loaded before **lwarp**.

```
191 \newcommand*\LWR@earlyloadnever[2]{%
192 \@ifpackageloaded{#1}{%
193 \PackageError{lwarp}
194 {Package #1 is not supported by lwarp's HTML conversion.
195 Package(s) #2 may be useful instead}
196 {Package #1 might conflict with lwarp in some way,
197 or is superceded by another package.
198 For a possible alternative, see package(s) #2.}
199 }{}%
200 }
```

25.2 Error for disallowed packages loaded before lwarp

```
201 \LWR@earlyloadnever{ae}{lmodern}
202 \LWR@earlyloadnever{aecc}{lmodern}
203 \LWR@earlyloadnever{boxedminipage}{boxedminipage2e}
204 \LWR@earlyloadnever{caption2}{caption}
205 % \LWR@earlyloadnever{ccaption}{caption}% might be preloaded by memoir
206 \LWR@earlyloadnever{fancyheadings}{fancyhdr}
207 \LWR@earlyloadnever{glossary}{glossaries}
208 \LWR@earlyloadnever{t1enc}{fontenc, inputenc, inputenx}
209 \LWR@earlyloadnever{wasysym}{textcomp, amssymb, amsfonts, mnsymbol, fdsymbol}
```

25.3 Enforcing package loading after lwarp

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

The following packages must be loaded after **lwarp**:

```
210 \LWR@loadafter{a4}
211 \LWR@loadafter{a4wide}
212 \LWR@loadafter{a5comb}
213 \LWR@notmemoirloadafter{abstract}
214 \LWR@loadafter{acro}
215 \LWR@loadafter{acronym}
216 \LWR@loadafter{adjmulticol}
217 \LWR@loadafter{addlines}
218 \LWR@loadafter{ae}
219 \LWR@loadafter{aecc}
220 \LWR@loadafter{afterpage}
221 \LWR@loadafter{algorithm2e}
222 \LWR@loadafter{algorithmicx}
223 \LWR@loadafter{alltt}
224 \LWR@loadafter{amsmath}
225 \LWR@loadafter{amsthm}
226 \LWR@loadafter{anonchap}
227 \LWR@loadafter{anysize}
228 \LWR@notmemoirloadafter{appendix}
229 \LWR@loadafter{arabicfront}
230 \LWR@notmemoirloadafter{array}
231 % \LWR@loadafter{atbegshi}% used by morewrites
232 \LWR@loadafter{authblk}
233 \LWR@loadafter{axodraw2}
234 \LWR@loadafter{backref}
235 \LWR@loadafter{balance}
236 \LWR@loadafter{bigdelim}
237 \LWR@loadafter{bigstrut}
238 \LWR@loadafter{blowup}
239 \LWR@loadafter{bookmark}
240 \LWR@notmemoirloadafter{booktabs}
241 \LWR@loadafter{boxedminipage}
242 \LWR@loadafter{boxedminipage2e}
243 \LWR@loadafter{breakurl}
244 \LWR@loadafter{bytefield}
245 \LWR@loadafter{cancel}
246 \LWR@loadafter{caption}
247 \LWR@loadafter{caption2}
248 \LWR@loadafter{ccaption}
249 \LWR@loadafter{changebar}
250 \LWR@notmemoirloadafter{change page}
251 \LWR@notmemoirloadafter{chn gpage}
252 \LWR@loadafter{chappg}
253 \LWR@loadafter{chapterbib}
254 \LWR@loadafter{chemfig}
255 \LWR@loadafter{chemformula}
256 \LWR@loadafter{chemgreek}
257 \LWR@loadafter{chemmacros}
```

```
258 \LWR@loadafter{chemnum}
259 \LWR@loadafter{cite}
260 \LWR@loadafter{color}
261 \LWR@loadafter{colortbl}
262 \LWR@loadafter{continue}
263 \LWR@notmemoirloadafter{crop}
264 \LWR@loadafter{cuted}
265 \LWR@loadafter{cutwin}
266 \LWR@loadafter{dblfloatfix}
267 \LWR@loadafter{dblfnote}
268 \LWR@notmemoirloadafter{dcolumn}
269 \LWR@loadafter{diagbox}
270 \LWR@loadafter{draftwatermark}
271 \LWR@loadafter{easy-todo}
272 \LWR@loadafter{ebook}
273 \LWR@loadafter{ellipsis}
274 \LWR@loadafter{emptypage}
275 \LWR@loadafter{endfloat}
276 \LWR@loadafter{endheads}
277 \LWR@loadafter{endnotes}
278 \LWR@notmemoirloadafter{enumerate}
279 \LWR@loadafter{enumitem}
280 \LWR@notmemoirloadafter{epigraph}
281 \LWR@loadafter{epstopdf}
282 \LWR@loadafter{epstopdf-base}
283 \LWR@loadafter{errata}
284 \LWR@loadafter{eso-pic}
285 \LWR@loadafter{everypage}
286 \LWR@loadafter{everyshi}
287 \LWR@loadafter{extramarks}
288 \LWR@loadafter{fancybox}
289 \LWR@loadafter{fancyhdr}
290 \LWR@loadafter{fancyheadings}
291 \LWR@loadafter{fancyref}
292 \LWR@loadafter{fancyvrb}
293 \LWR@loadafter{figcaps}
294 \LWR@loadafter{figsize}
295 \LWR@loadafter{fix2col}
296 \LWR@loadafter{fixme}
297 \LWR@loadafter{fixmetodonotes}
298 \LWR@loadafter{flafter}
299 \LWR@loadafter{float}
300 \LWR@loadafter{floatflt}
301 \LWR@loadafter{floatpag}
302 \LWR@loadafter{floatrow}
303 \LWR@loadafter{fltrace}
304 \LWR@loadafter{flushend}
305 \LWR@loadafter{fncychap}
306 \LWR@loadafter{fnlineno}
307 \LWR@loadafter{fnpos}
```

```
308 % fontenc must be loaded before lwarp
309 % fontspec must be loaded before lwarp
310 \LWR@loadafter{footmisc}
311 \LWR@loadafter{footnote}
312 \LWR@loadafter{footnotehyper}
313 \LWR@loadafter{footnpag}
314 \LWR@loadafter{forest}
315 \LWR@loadafter{framed}
316 \LWR@loadafter{ftnright}
317 \LWR@loadafter{fullpage}
318 \LWR@loadafter{fullwidth}
319 \LWR@loadafter{fwlw}
320 \LWR@loadafter{geometry}
321 \LWR@loadafter{glossaries}
322 % \LWR@loadafter{graphics}% pre-loaded by xunicode
323 % \LWR@loadafter{graphicx}% pre-loaded by xunicode
324 \LWR@loadafter{glossary}
325 \LWR@loadafter{grffile}
326 \LWR@loadafter{grid}
327 \LWR@loadafter{gridset}
328 \LWR@loadafter{hang}
329 \LWR@loadafter{hanging}
330 \LWR@loadafter{hypcap}
331 \LWR@loadafter{hypdestopt}
332 \LWR@loadafter{hypernat}
333 \LWR@loadafter{hyperref}
334 \LWR@loadafter{hyperxmp}
335 \LWR@loadafter{hyphenat}
336 \LWR@loadafter{idxlayout}
337 \LWR@loadafter{ifoddpag}
338 \LWR@loadafter{indentfirst}
339 % inputenc must be loaded before lwarp
340 % inputenx must be loaded before lwarp
341 \LWR@loadafter{keyfloat}
342 \LWR@loadafter{layout}
343 \LWR@loadafter{letterspace}
344 \LWR@loadafter{lettrine}
345 \LWR@loadafter{lineno}
346 \LWR@loadafter{lips}
347 \LWR@loadafter{listings}
348 \LWR@loadafter{longtable}
349 \LWR@loadafter{lscap}
350 \LWR@loadafter{ltablex}
351 \LWR@loadafter{ltcaption}
352 \LWR@loadafter{ltxgrid}
353 \LWR@loadafter{ltxtable}
354 \LWR@loadafter{lua-check-hyphen}
355 \LWR@loadafter{luacolor}
356 \LWR@loadafter{luatodonotes}
357 \LWR@loadafter{magaz}
```

```
358 \LWR@loadafter{marginfit}
359 \LWR@loadafter{marginfix}
360 \LWR@loadafter{marginnote}
361 \LWR@loadafter{mcaption}
362 \LWR@loadafter{mdframed}
363 \LWR@loadafter{memhfixc}
364 \LWR@loadafter{metalogo}
365 \LWR@loadafter{mhchem}
366 \LWR@loadafter{microtype}
367 \LWR@loadafter{midfloat}
368 \LWR@loadafter{midpage}
369 \LWR@loadafter{morefloats}
370 \LWR@notmemoirloadafter{moreverb}
371 % morewrites must be loaded before lwarp
372 \LWR@notmemoirloadafter{mparhack}
373 %\LWR@loadafter{multicol}% loaded by ltxdoc
374 \LWR@loadafter{multirow}
375 \LWR@loadafter{multitoc}
376 \LWR@loadafter{nameref}
377 \LWR@loadafter{natbib}
378 \LWR@notmemoirloadafter{needspace}
379 % newclude must be loaded before lwarp
380 \LWR@loadafter{newtxmath}
381 % newunicodechar must be loaded before lwarp
382 \LWR@notmemoirloadafter{nextpage}
383 \LWR@loadafter{nicefrac}
384 \LWR@loadafter{nonfloat}
385 \LWR@loadafter{nonumonpart}
386 \LWR@loadafter{nopageno}
387 \LWR@loadafter{nowidow}
388 \LWR@loadafter{ntheorem}
389 \LWR@loadafter{overpic}
390 \LWR@loadafter{pagegrid}
391 \LWR@notmemoirloadafter{pagenote}
392 \LWR@loadafter{pagesel}
393 \LWR@loadafter{paralist}
394 \LWR@loadafter{parnotes}
395 \LWR@notmemoirloadafter{parskip}
396 \LWR@loadafter{pbox}
397 \LWR@loadafter{pdfrender}
398 \LWR@loadafter{pdfscape}
399 \LWR@loadafter{pdfsync}
400 \LWR@loadafter{pfnote}
401 \LWR@loadafter{phfqit}
402 \LWR@loadafter{placeins}
403 \LWR@loadafter{prelim2e}
404 \LWR@loadafter{prettyref}
405 \LWR@loadafter{preview}
406 \LWR@loadafter{quotchap}
407 \LWR@loadafter{quoting}
```

```
408 \LWR@loadafter{ragged2e}
409 \LWR@loadafter{realscripts}
410 \LWR@loadafter{register}
411 \LWR@loadafter{relsize}
412 \LWR@loadafter{resizegather}
413 \LWR@loadafter{romanbar}
414 \LWR@loadafter{romanbarpagenumber}
415 \LWR@loadafter{rotating}
416 \LWR@loadafter{rotfloat}
417 \LWR@loadafter{savetrees}
418 % \LWR@loadafter{scalegnt}% loaded by babel-french
419 \LWR@loadafter{schemata}
420 \LWR@loadafter{scrextend}
421 \LWR@loadafter{scrhack}
422 \LWR@loadafter{scrlayer}
423 \LWR@loadafter{scrlayer-notecolumn}
424 \LWR@loadafter{scrlayer-scrpage}
425 \LWR@loadafter{section}
426 \LWR@loadafter{sectionbreak}
427 \LWR@loadafter{sectsty}
428 \LWR@notmemoirloadafter{setspace}
429 \LWR@loadafter{shadow}
430 \LWR@notmemoirloadafter{showidx}
431 \LWR@loadafter{showkeys}
432 \LWR@loadafter{sidecap}
433 \LWR@loadafter{sidenotes}
434 \LWR@loadafter{siunitx}
435 \LWR@loadafter{soul}
436 \LWR@loadafter{soulpos}
437 \LWR@loadafter{soulutf8}
438 \LWR@loadafter{stabular}
439 \LWR@loadafter{stfloats}
440 \LWR@loadafter{subfig}
441 \LWR@loadafter{subfigure}
442 \LWR@loadafter{supertabular}
443 \LWR@loadafter{t1inc}
444 \LWR@loadafter{tabls}
445 \LWR@notmemoirloadafter{tabularx}
446 \LWR@loadafter{tabulary}
447 \LWR@loadafter{textarea}
448 % \LWR@loadafter{textcomp}% maybe before lwarp with font packages
449 \LWR@loadafter{textfit}
450 \LWR@loadafter{textpos}
451 \LWR@loadafter{theorem}
452 \LWR@loadafter{threeparttable}
453 \LWR@loadafter{tikz}
454 \LWR@loadafter{titleps}
455 \LWR@loadafter{titlesec}
456 \LWR@loadafter{titletoc}
457 \LWR@notmemoirloadafter{titling}
```

```

458 % \LWR@loadafter{tocbasic}% preloaded by koma-script classes
459 \LWR@notmemoirloadafter{tocbibind}
460 \LWR@loadafter{toccenter}
461 \LWR@notmemoirloadafter{tocloft}
462 \LWR@loadafter{tocstyle}
463 \LWR@loadafter{todo}
464 \LWR@loadafter{todonotes}
465 \LWR@loadafter{transparent}
466 \LWR@loadafter{trimclip}
467 \LWR@loadafter{trivfloat}
468 \LWR@loadafter{turnthepage}

469 % \LWR@loadafter{typearea}% preloaded by koma-script classes
470 \LWR@loadafter{ulem}
471 \LWR@loadafter{underscore}
472 \LWR@loadafter{upref}
473 \LWR@loadafter{url}
474 \LWR@loadafter{varioref}% no lwarp package provided
475 \LWR@notmemoirloadafter{verse}
476 \LWR@loadafter{vertbars}
477 \LWR@loadafter{vmargin}
478 \LWR@loadafter{vowel}
479 \LWR@loadafter{vwcol}
480 \LWR@loadafter{wallpaper}
481 \LWR@loadafter{wasysym}
482 \LWR@loadafter{watermark}
483 \LWR@loadafter{wrapfig}
484 \LWR@loadafter{xcolor}
485 \LWR@loadafter{xellipsis}
486 \LWR@loadafter{xfrac}
487 \LWR@loadafter{xltabular}
488 \LWR@loadafter{xltxtra}
489 \LWR@loadafter{xmpincl}
490 \LWR@loadafter{xpiano}
491 \LWR@loadafter{xtab}
492 \LWR@loadafter{xurl}
493 \LWR@loadafter{xy}
494 \LWR@loadafter{zwpgelayout}

```

26 Required packages

These packages are automatically loaded by **lwarp** when generating HTML output. Some of them are also automatically loaded when generating print output, but some are not.

for HTML output: 495 \begin{warpHTML}

Load **fontspec** if necessary:

```
496 \ifxetexorluatex
497 \@ifpackageloaded{fontspec}{\}{
498 \usepackage[no-math]{fontspec}
499 }
```

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

```
500 \defaultfontfeatures[\rmfamily]{Ligatures={NoCommon,TeX}}
501 \defaultfontfeatures[\sffamily]{Ligatures={NoCommon,TeX}}
502 \defaultfontfeatures[\ttfamily]{Ligatures=NoCommon}
503 \else
```

pdflatex only: Only pre-loaded if **pdflatex** is being used.

Pkg microtype

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

```
504 \RequirePackage {microtype}
505
506 \microtypesetup{
507   protrusion=false,
508   expansion=false,
509   tracking=false,
510   kerning=false,
511   spacing=false}
512
513 \DisableLigatures[f,q,t,T,Q]{encoding = *,family = *}
514 \fi
515 \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: 516 \begin{warpHTML}
 517 \RequirePackage[paperheight=190in,paperwidth=20in,%
 518 left=2in,right=6in,%
 519 top=1in,bottom=1in,%
 520]{geometry}
 521 \@twosidefalse
 522 \@mparswitchfalse
 523 \end{warpHTML}

for HTML & PRINT: 524 \begin{warpall}

Pkg xparse

\LaTeX 3 command argument parsing

525 \RequirePackage{xparse}

526 \end{warpall}

for HTML output: 527 \begin{warpHTML}

Pkg expl3

\LaTeX 3 programming

528 \RequirePackage{expl3}

Pkg gettitlestring

Used to emulate \nameref.

529 \RequirePackage{gettitlestring}

Pkg everyhook

everyhook is used to patch paragraph handling.

530 \RequirePackage{everyhook}

531 \end{warpHTML}

for HTML & PRINT: 532 \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>

```

533 \RequirePackage{filecontents}
534
535 \@ifpackagelater{filecontents}{2011/10/09}%
536 {}
537 {
538 \newwrite\fcwrite
539 \let\LWR@origfilec@ntents\filec@ntents
540 \def\filec@ntents{\def\chardef##1\write{\let\reserved@c\fcwrite}\LWR@origfilec@ntents}
541 }

542 \end{warpall}

```

for HTML output: 543 \begin{warpHTML}

Pkg xifthen

```
544 \RequirePackage{xifthen}
```

Pkg xstring

```
545 \RequirePackage{xstring}
```

Pkg verbatim

```
546 \RequirePackage{verbatim}
```

Pkg makeidx

```
547 \RequirePackage{makeidx}
548 \makeindex
```

Pkg calc

```
549 \RequirePackage{calc}
```

Pkg refcount

Provides \setcounterref, \setcounterpageref, etc.

```
550 \RequirePackage{refcount}
```

Pkg newfloat

```
551 \RequirePackage{newfloat}
```

```
552 \end{warpHTML}
```

for HTML & PRINT: 553 \begin{warpall}

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

```
554 \RequirePackage{environ}
```

```
555 \end{warpall}
```

for HTML output: 556 \begin{warpHTML}

Pkg zref Used for cross-references.

```
557 \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 option `leqno` is passed beforehand to be picked up both here and by **newtxmath** if it is used.

```
558 % \PassOptionsToPackage{leqno}{amsmath}% disabled to test centered display math
559 \RequirePackage{amsmath}
```

Patches to allow `\eqref` inside a caption:

```
560 \def\maketag@@@#1{\text{#1}}
561 \def\tagform@#1{\maketag@@@(\ignorespaces#1\unskip)}
```

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

```
562 \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.

```
563 \newrobustcmd*{\LWR@printlength}[1]{%
564 \begingroup%
565 \uselengthunit{PT}%
566 \renewcommand*{\unitspace}{}%
567 \ifdimless{#1}{10pt}{%
568   \printlength{#1}%
```

```

569 }{%
570     \rndprintlength{#1}%
571 }%
572 \endgroup%
573 }

574 \end{warpHTML}

```

for PRINT output: 575 \begin{warpprint}

Pkg varwidth Used for print-mode lateximage:

```

576 \RequirePackage{varwidth}

577 \end{warpprint}

```

27 Loading packages

for HTML output: 578 \begin{warpHTML}

Remember the original \RequirePackage:

```

579 \LetLtxMacro\LWR@origRequirePackage\RequirePackage

```

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

```

580 \newcommand*\LWR@requirepackagenames{}

```

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

```

581 \newcommand*\LWR@parsedrequirepackagenames{}

```

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

Note that argument 4 is passed directly to \StrBetween.

```

582 \newcommand*\LWR@findword[3][,]{%
583     \StrBetween[#3,\numexpr#3+1]{#1#2#1}{#1}{#1}%
584 }

```

\LWR@lookforpackagename {*index*}

If this is an **lwarp**-supported package name, re-direct it to the **lwarp** version by renaming it **lwarp-** followed by the original name.

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

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

Find the index'th package name from the list:

```
586 \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`.

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

See if the package name was found:

```
588 \IfStrEq{\LWR@strresulttwo}{}%
589 {%
590 }% no filename
591 {% yes filename
```

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

```
592   \IfFileExists{lwarp-\LWR@strresulttwo.sty}%
593   {% lwarp-* file found
594     \ifdefvoid{\LWR@parsedrequirepackagenames}{%
595       \edef\LWR@parsedrequirepackagenames{lwarp-\LWR@strresulttwo}%
596     }{%
597       \edef\LWR@parsedrequirepackagenames{%
598         \LWR@parsedrequirepackagenames,lwarp-\LWR@strresulttwo%
599       }%
600     }%
601   }%
602   {%

603     \ifdefvoid{\LWR@parsedrequirepackagenames}{%
604       \edef\LWR@parsedrequirepackagenames{\LWR@strresulttwo}%
605     }{%
606       \edef\LWR@parsedrequirepackagenames{%
607         \LWR@parsedrequirepackagenames,\LWR@strresulttwo%
608       }%
609     }%
610   }% no lwarp-* file
611 }% yes filename
612 }
```

`\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.

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

Redirect up to nine names:

```
614 \renewcommand*{\LWR@requirepackagenames}{#2}%
615 \renewcommand*{\LWR@parsedrequirepackagenames}{}%
616 \LWR@lookforpackagename{1}%
617 \LWR@lookforpackagename{2}%
618 \LWR@lookforpackagename{3}%
619 \LWR@lookforpackagename{4}%
620 \LWR@lookforpackagename{5}%
621 \LWR@lookforpackagename{6}%
622 \LWR@lookforpackagename{7}%
623 \LWR@lookforpackagename{8}%
624 \LWR@lookforpackagename{9}%
```

`\RequirePackage` depending on the options and version:

```
625 \IfValueTF{#1}%
626 {% options given
627   \IfValueTF{#3}% version given?
628   {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}[#3]}%
629   {\LWR@origRequirePackage[#1]{\LWR@parsedrequirepackagenames}}%
630 }%
631 {% no options given
632   \IfValueTF{#3}% version given?
633   {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}[#3]}%
634   {\LWR@origRequirePackage{\LWR@parsedrequirepackagenames}}%
635 }%
636 }
637 \LetLtxMacro\usepackage\RequirePackage
```

`\LWR@ProvidesPackagePass` {*<pkgname>*} [*<version>*]

Uses the original package, including options.

```
638 \NewDocumentCommand{\LWR@ProvidesPackagePass}{m o}{
639 \PackageInfo{lwarp}{Using package ‘#1’ and adding lwarp modifications, including options,}%
640 \IfValueTF{#2}
641 {\ProvidesPackage{lwarp-#1}[#2]}
642 {\ProvidesPackage{lwarp-#1}}
643 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{#1}}
644 \ProcessOptions\relax
```

```

645 \IfValueTF{#2}
646 {\LWR@origRequirePackage{#1}[#2]}
647 {\LWR@origRequirePackage{#1}}
648 }

```

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

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

```

649 \NewDocumentCommand{\LWR@ProvidesPackageDrop}{m o}{
650 \PackageInfo{lwarp}{Replacing package ‘#1’ with the lwarp version, discarding options,}%
651 \IfValueTF{#2}
652 {\ProvidesPackage{lwarp-#1}[#2]}
653 {\ProvidesPackage{lwarp-#1}}

```

Ignore all options.

```

654 \DeclareOption*{}

```

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

```

655 % \ProcessOptions\relax% original LaTeX code
656 \let\ds@\empty%      from the original \ProcessOptions
657 \edef\@curroptions{}% lwarp modification to \ProcessOptions
658 \@processoptions\relax% from the original \ProcessOptions
659 }

```

```

660 \end{warpHTML}

```

28 Additional required packages

for HTML output: 661 `\begin{warpHTML}`

`Pkg` `caption`

```

662 \RequirePackage{caption}%

```

```

663 \end{warpHTML}

```

29 File handles

Defines file handles for writes.

for HTML & PRINT: 664 `\begin{warpall}`

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

665 `\newwrite\LWR@quickfile%`

666 `\end{warpall}`

for HTML output: 667 `\begin{warpHTML}`

`\LWR@lateximagesfile` For lateximages.txt.

668 `\newwrite\LWR@lateximagesfile`

669 `\end{warpHTML}`

30 Include a file

During HTML output, `\include{<filename>}` causes the following to occur:

1. **lwarp** creates `<filename>_html_inc.tex` whose contents are:
`\input <filename>.tex`
2. `<filename>_html_inc.tex` is then `\included` instead of `<filename>.tex`.
3. `<filename>_html_inc.aux` is automatically generated and used by \TeX .

for HTML output: 670 `\begin{warpHTML}`

`\include` `{\filename}`

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

671 `\def\@include#1 {%`

672 `\immediate\openout\LWR@quickfile #1_html_inc.tex% lwarp`

673 `\immediate\write\LWR@quickfile{\string\input{#1.tex}}% lwarp`

```

674 \immediate\closeout\LWR@quickfile% lwarp
675 \LWR@origclearpage% \changed
676 \if@filesw
677     \immediate\write\@mainaux{\string\@input{#1_html_inc.aux}}% changed
678 \fi
679 \@tempswatrue
680 \if@partsw
681     \@tempswafalse
682     \edef\reserved@a{\@partlist\do
683         {\ifx\reserved@a\reserved@b\@tempswatrue\fi}%
684     \fi
685 \if@tempswa
686     \let\@auxout\@partaux
687     \if@filesw
688         \immediate\openout\@partaux #1_html_inc.aux % changed
689         \immediate\write\@partaux{\relax}%
690     \fi
691     \@input{#1_html_inc.tex}% changed
692     \LWR@origclearpage% changed
693     \@writeckpt{#1}%
694     \if@filesw
695         \immediate\closeout\@partaux
696     \fi
697 \else
698     \deadcycles\z@
699     \@nameuse{cp@#1}%
700 \fi
701 \let\@auxout\@mainaux%
702 }
703 }

704 \end{warpHTML}

```

31 Copying a file

for HTML output: 705 \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.

```

706 \newwrite\LWR@copyoutfile % open the file to write to
707 \newread\LWR@copyinfile % open the file to read from
708
709 \newcommand*\LWR@copyfile[2]{%

```

```

710 \LWR@traceinfo{LWR@copyfile: copying #1 to #2}
711
712 \immediate\openout\LWR@copyoutfile=#2
713 \openin\LWR@copyinfile=#1
714 \begingroup\endlinechar=-1
715 \makeatletter
716
717 \LWR@traceinfo{LWR@copyfile: about to loop}
718
719 \loop\unless\ifeof\LWR@copyinfile
720   \LWR@traceinfo{LWR@copyfile: one line}
721   \read\LWR@copyinfile to\LWR@fileline % Read one line and store it into \LWR@fileline
722 %   \LWR@fileline\par % print the content into the pdf
723 % print the content:
724   \immediate\write\LWR@copyoutfile{\unexpanded\expandafter{\LWR@fileline}}%
725 \repeat
726 \immediate\closeout\LWR@copyoutfile
727 \LWR@traceinfo{LWR@copyfile: done}
728 \endgroup
729 }

730 \end{warpHTML}

```

32 Debugging messages

for HTML & PRINT: 731 \begin{warpall}

Bool LWR@tracinglwarp True if tracing is turned on.

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

\tracinglwarp Turns on the debug tracing messages.

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

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

```

734 \newcommand{\LWR@traceinfo}[1]{%
735 \ifbool{LWR@tracinglwarp}%
736 {%
737   \typeout{*** lwarp: #1}%
738 }%
739 {}%
740 }

```

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

Default: false

```
741 \newbool{HTMLDebugComments}
742 \boolfalse{HTMLDebugComments}
```

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

```
743 \AfterEndPreamble{
744 \LWR@traceinfo{AfterEndPreamble}
745 }
746
747 \AtBeginDocument{
748 \LWR@traceinfo{AtBeginDocument}
749 }

750 \end{warpall}
```

33 Defining print and HTML versions of macros and environments

The following refers to defining objects inside **lwarp**, and is not for the user's document.

Many macros and environments must be provided as both print and HTML versions.

While generating the print version of a document, the original macros as defined by \TeX and its packages are used as-is.

While generating the HTML version of a document, the original macro or environment is redefined to call a new HTML version or a copy of the original print version. The new HTML versions of macros and environments are used most of the time. Copies of the print versions are used inside a `lateximage` environment, which draws and remembers an image of the printed output, and also several other places.

The general structure for providing print and HTML versions of a macro or environment is as follows:

For a preexisting macro, not defined with `xparse`: An HTML version is provided with a special name, inside a `warpHTML` environment, then `\LWR@formatted` is used to redefine and patch various macros:

```
\begin{warpHTML}
\newcommand{\LWR@HTML@name}{...}% may also use xparse
```

```
\LWR@formatted{name}
\end{warpHTML}
```

`\LWR@formatted{name}` copies the original print version, then redefines `\name` to use either the print or HTML version depending on which mode **lwarp** is using. **xparse** may be used to define the new HTML version, even if the original did not use **xparse**.

For a preexisting environment, not defined with xparse: The process is similar. Note the use of `\LWR@formattedenv` instead of `\LWR@formatted`.

```
\begin{warpHTML}
\newenvironment{LWR@HTML@name}{...}% may also use xparse

\LWR@formattedenv{name}
\end{warpHTML}
```

If the original used xparse: A copy must be made using a new name:

```
\begin{warpHTML}
\NewDocumentCommand{\LWR@print@name}{..}{..}% copy the original

\NewDocumentCommand{\LWR@HTML@name}{..}{..}% or use \newcommand

\LWR@formatted{name}
\end{warpHTML}
```

Similar for an environment, using `\LWR@formattedenv`. (`\LWR@formatted` and `\LWR@formattedenv` use `\LetLtxMacro` to copy the original print definition, which may not work with macros and environments created by **xparse**, so the print version must be manually recreated in the **lwarp** source.)

For a new macro or environment, not using xparse for the print version:

```
\begin{warpall}
\newcommand{name}{...}% NOT xparse!
\end{warpall}

\begin{warpHTML}
\newcommand{\LWR@HTML@name}{...}% may use xparse for HTML

\LWR@formatted{name}
\end{warpHTML}
```

Similar for an environment. The plain `\name` or environment name is used for the printed version, and is placed inside `warpall`. **xparse** may be used for the `\LWR@HTML@<name>` version.

For a new macro or environment, using xparse: It is possible to use **xparse** for an entirely new macro or environment by defining the `\LWR@print@<name>` version with **xparse**, along with `\name` defined without **xparse** to refer directly to the `\LWR@print` version:

```

\begin{warpall}
\NewDocumentCommand{\LWR@print@name}{...} {...}%      -or-
\NewDocumentEnvironment{\LWR@print@name}{...} {...} {...}

% Simply a call to \LWR@print@name:
\newcommand{\name}{\LWR@print@name}%                  -or-
\newenvironment{name}{\LWR@print@name}{\endLWR@print@name}
\end{warpall}

\begin{warpHTML}
\NewDocumentCommand{\LWR@HTML@name}{...} {...}%      -or-
\NewDocumentEnvironment{\LWR@HTML@name}{...} {...} {...}

\LWR@formatted{name}%      -or-
\LWR@formattedenv{name}
\end{warpHTML}

```

In general, `\LWR@formatted` or `\LWR@formattedenv` are placed inside a `warpHTML` environment, and while producing an HTML document they do the following:

- Macros are modified:
 1. The pre-existing print version `\name` is saved as `\LWR@print@<name>`, unless `\LWR@print@<name>` is already defined.
 2. The original `\name` is redefined to call either the print or HTML version depending on which format is in use at the moment, as set by `\LWR@formatting`, which is defined as either “print” or “HTML”.
- When **lwarp** is producing a print document, the original definitions are used, as well as any new definitions defined in `warpall` above.
- When **lwarp** is generating HTML output, `\LWR@formatting` is set to “HTML”, and `\name` is directed to `\LWR@HTML@<name>`.
- When **lwarp** is generating HTML output but enters a `lateximage` environment, or for some other reason needs to draw images using the original print definitions, `\LWR@formatting` is changed to “print” and `\name` is then redirected to `\LWR@print@<name>`, which was the original `\name`.

Since arguments are not handled by the new `\name`, any star and other arguments are processed by the print or HTML version.

Expandable versions are also provided as well:

```
\LWR@expandableformatted
\LWR@expandableformattedenv
```

(Older versions of **lwarp** used `\LetLtxMacro` for everything, but this could fail when using macros defined by **xparse**. This older system is still in use for many definitions.)

for HTML output: 751 `\begin{warpHTML}`

`\LWR@formatting` Remembers if selected print/HTML formatting.

Used while `\LWR@restoreorigformatting`, such as in an `lateximage`. May be set to either “print” or “HTML”.

```
752 \newcommand*{\LWR@formatting}{HTML}
```

`\LWR@formatted` `{\macroname}` No backslash in the macro name.

If not yet defined, defines `\LWR@print@<name>` as the original print-mode `\<name>`. Also redefines `\<name>` to use `\LWR@<format>@<name>`, where `<format>` is set by `\LWR@formatting`, and is print or HTML.

```
753 \newcommand*{\LWR@formatted}[1]{%
754   \ifcsundef{\LWR@print@#1}{%
755     \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
756     \csname#1\endcsname%
757   }{}%
758   \ifcsundef{#1}{%
759     \expandafter\newrobustcmd\csname #1\endcsname{%
760       \csuse{\LWR@\LWR@formatting @#1}%
761     }%
762   }{%
763     \expandafter\renewrobustcmd\csname #1\endcsname{%
764       \csuse{\LWR@\LWR@formatting @#1}%
765     }%
766   }%
767 }
```

`\LWR@expandableformatted` `{\macroname}` No backslash in the macro name.

An expandable version of `\LWR@formatted`.

```
768 \newcommand*{\LWR@expandableformatted}[1]{%
769   \ifcsundef{\LWR@print@#1}{%
```

```

770      \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
771      \csname#1\endcsname%
772    }{}%
773    \ifcsundef{#1}{%
774      \expandafter\newcommand\csname #1\endcsname{%
775        \csuse{LWR@LWR@formatting @#1}%
776      }%
777    }{%
778      \expandafter\renewcommand\csname #1\endcsname{%
779        \csuse{LWR@LWR@formatting @#1}%
780      }%
781    }%
782 }

```

`\LWR@formattedenv` $\{\langle environmentname \rangle\}$

If not yet defined, defines the environment `LWR@print@<name>` as the original print-mode `<name>`. Also redefines the environment `<name>` to use environment `LWR@<format>@<name>`, where `<format>` is set by `\LWR@formatting`, and is print or HTML.

```

783 \newcommand*{\LWR@formattedenv}[1]{%
784   \ifcsundef{LWR@print@#1}{%
785     \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
786     \csname#1\endcsname%
787     \csletcs{endLWR@print@#1}{end#1}%
788   }{}%
789   \DeclareDocumentEnvironment{#1}{}%
790   {\csuse{LWR@LWR@formatting @#1}}%
791   {\csuse{endLWR@LWR@formatting @#1}}%
792 }

```

`\LWR@expandableformattedenv` $\{\langle environmentname \rangle\}$

An expandable version of `LWR@formattedenv`.

```

793 \newcommand*{\LWR@expandableformattedenv}[1]{%
794   \ifcsundef{LWR@print@#1}{%
795     \expandafter\LetLtxMacro\csname LWR@print@#1\expandafter\endcsname%
796     \csname#1\endcsname%
797     \csletcs{endLWR@print@#1}{end#1}%
798   }{}%
799   \DeclareExpandableDocumentEnvironment{#1}{}%
800   {\csuse{LWR@LWR@formatting @#1}}%
801   {\csuse{endLWR@LWR@formatting @#1}}%
802 }
803 \end{warpHTML}

```

34 HTML-conversion output modifications

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

for HTML & PRINT: 804 \begin{warpall}

34.1 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

```
805 \newbool{FormatEPUB}
806 \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

```
807 \newbool{FormatWP}
808 \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.¹⁶

```
809 \newbool{WPMarkFloats}
810 \boolfalse{WPMarkFloats}
```

Bool **WPMarkMinipages** Adds

Default: false

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

¹⁶Perhaps some day word processors will have HTML import options for identifying <figure> and <figcaption> tags for figures and tables.

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

```
811 \newbool{WPMarkMinipages}
812 \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.

```
813 \newbool{WPMarkTOC}
814 \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.

```
815 \newbool{WPMarkLOFT}
816 \boolfalse{WPMarkLOFT}
```

Bool 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.

```
817 \newbool{WPMarkMath}
818 \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**.

See table 6 on page 148.

```
819 \newbool{WPTitleHeading}
820 \boolfalse{WPTitleHeading}
```

```
821 \end{warpall}
```

34.2 Heading adjustments

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

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

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

for HTML output: 822 \begin{warpHTML}

```
823 \AtBeginDocument{
824 \ifbool{FormatWP}{
825 \@ifundefined{chapter}{
826 \ifbool{WPTitleHeading}{% part and section starting at h2
827 \renewcommand*\LWR@tagtitle}{h1}
828 \renewcommand*\LWR@tagtitleend}{/h1}
829 \renewcommand*\LWR@tagpart}{h2}
830 \renewcommand*\LWR@tagpartend}{/h2}
831 \renewcommand*\LWR@tagsection}{h3}
832 \renewcommand*\LWR@tagsectionend}{/h3}
833 \renewcommand*\LWR@tagsubsection}{h4}
834 \renewcommand*\LWR@tagsubsectionend}{/h4}
835 \renewcommand*\LWR@tagsubsubsection}{h5}
836 \renewcommand*\LWR@tagsubsubsectionend}{/h5}
837 \renewcommand*\LWR@tagparagraph}{h6}
838 \renewcommand*\LWR@tagparagraphend}{/h6}
839 \renewcommand*\LWR@tagsubparagraph}{span class="subparagraph"}
840 \renewcommand*\LWR@tagsubparagraphend}{/span}
841 }% WPTitleHeading
842 {% not WPTitleHeading, part and section starting at h1
843 \renewcommand*\LWR@tagtitle}{div class="title"}
844 \renewcommand*\LWR@tagtitleend}{/div}
845 \renewcommand*\LWR@tagpart}{h1}
846 \renewcommand*\LWR@tagpartend}{/h1}
847 \renewcommand*\LWR@tagsection}{h2}
848 \renewcommand*\LWR@tagsectionend}{/h2}
849 \renewcommand*\LWR@tagsubsection}{h3}
850 \renewcommand*\LWR@tagsubsectionend}{/h3}
851 \renewcommand*\LWR@tagsubsubsection}{h4}
852 \renewcommand*\LWR@tagsubsubsectionend}{/h4}
853 \renewcommand*\LWR@tagparagraph}{h5}
854 \renewcommand*\LWR@tagparagraphend}{/h5}
855 \renewcommand*\LWR@tagsubparagraph}{h6}
856 \renewcommand*\LWR@tagsubparagraphend}{/h6}
857 }% not WPTitleHeading
858 }% chapter undefined
```

```

859 {% chapter defined
860 \ifbool{WPTitleHeading}{%
861 {% not WPTitleHeading, part and chapter starting at h1
862 \renewcommand*{\LWR@tagtitle}{div class="title"}
863 \renewcommand*{\LWR@tagtitleend}{/div}
864 \renewcommand*{\LWR@tagpart}{h1}
865 \renewcommand*{\LWR@tagpartend}{/h1}
866 \renewcommand*{\LWR@tagchapter}{h2}
867 \renewcommand*{\LWR@tagchapterend}{/h2}
868 \renewcommand*{\LWR@tagsection}{h3}
869 \renewcommand*{\LWR@tagsectionend}{/h3}
870 \renewcommand*{\LWR@tagsubsection}{h4}
871 \renewcommand*{\LWR@tagsubsectionend}{/h4}
872 \renewcommand*{\LWR@tagsubsubsection}{h5}
873 \renewcommand*{\LWR@tagsubsubsectionend}{/h5}
874 \renewcommand*{\LWR@tagparagraph}{h6}
875 \renewcommand*{\LWR@tagparagraphend}{/h6}
876 \renewcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
877 \renewcommand*{\LWR@tagsubparagraphend}{/span}
878 }% not WPTitleHeading
879 }% chapter defined
880 }{}% FormatWP
881 }% AtBeginDocument

882 \end{warpHTML}

```

35 Remembering original formatting macros

for HTML output: 883 \begin{warpHTML}

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

Some packages redefine \#, which is used to generate HTML, so the original must be remembered here.

```

884 \chardef\LWR@origpound='\#

885 \let\LWR@origcomma\,
886 \let\LWR@origthinspace\thinspace
887 \let\LWR@origtilde~
888 \let\LWR@origenskip\enskip
889 \let\LWR@origquad\quad
890 \let\LWR@origqqquad\qqquad
891 \let\LWR@orighfill\hfill

```

```
892 \let\LWR@orighfil\hfil
893 \let\LWR@orighss\hss
894 \let\LWR@origllap\llap
895 \let\LWR@origrlap\rlap
896 \let\LWR@orighfilneg\hfilneg
897
898 \let\LWR@orighrulefill\hrulefill
899 \let\LWR@origdotfill\dotfill
900
901 \let\LWR@origmedskip\medskip
902 \let\LWR@origbigskip\bigskip
903
904 \let\LWR@origtextellipsis\textellipsis
905
906 \let\LWR@orignormalsize\normalsize
907 \let\LWR@origsmall\small
908 \let\LWR@origfootnotesize\footnotesize
909 \let\LWR@origscriptsize\scriptsize
910 \let\LWR@origtiny\tiny
911 \let\LWR@origlarge\large
912 \let\LWR@origLarge\Large
913 \let\LWR@origLARGE\LARGE
914 \let\LWR@orighuge\huge
915 \let\LWR@origHuge\Huge
916
917 \LetLtxMacro\LWR@origtextrm\textrm
918 \LetLtxMacro\LWR@origtextsf\textsf
919 \LetLtxMacro\LWR@origtexttt\texttt
920 \LetLtxMacro\LWR@origtextnormal\textnormal
921 \LetLtxMacro\LWR@origtextbf\textbf
922 \LetLtxMacro\LWR@origtextmd\textmd
923 \LetLtxMacro\LWR@origtextit\textit
924 \LetLtxMacro\LWR@origtextsl\textsl
925 \LetLtxMacro\LWR@origtextsc\textsc
926 \LetLtxMacro\LWR@origtextup\textup
927 \LetLtxMacro\LWR@origemph\emph
928
929 \LetLtxMacro\LWR@origrmfamily\rmfamily
930 \LetLtxMacro\LWR@origsffamily\sffamily
931 \LetLtxMacro\LWR@origttfamily\ttfamily
932 \LetLtxMacro\LWR@origbfseries\bfseries
933 \LetLtxMacro\LWR@origmdseries\mdseries
934 \LetLtxMacro\LWR@origupshape\upshape
935 \LetLtxMacro\LWR@origslshape\slshape
936 \LetLtxMacro\LWR@origscshape\scshape
937 \LetLtxMacro\LWR@origitshape\itshape
938 \LetLtxMacro\LWR@origem\em
939 \LetLtxMacro\LWR@orignormalfont\normalfont
940
941 \let\LWR@origraggedright\raggedright
```

```
942 \let\LWR@origonecolumn\onecolumn
943
944 \let\LWR@origsp\sp
945 \let\LWR@origsb\sb
946 \LetLtxMacro\LWR@origtextsuperscript\textsuperscript
947 \LetLtxMacro\LWR@orig@textsuperscript\@textsuperscript
948
949 \AtBeginDocument{
950 \LetLtxMacro\LWR@origtextsubscript\textsubscript
951 \LetLtxMacro\LWR@orig@textsubscript\@textsubscript
952 }
953
954 \LetLtxMacro\LWR@origunderline\underline


955 \let\LWR@origraggedright\raggedright
956 \let\LWR@origraggedleft\raggedleft
957 \let\LWR@origcentering\centering


958 \let\LWR@orignewpage\newpage
959
960 \let\LWR@origpagestyle\pagestyle
961 \let\LWR@origthispagestyle\thispagestyle
962 \LetLtxMacro\LWR@origpagenumbering\pagenumbering
963
964 \let\LWR@orignewline\newline
965
966
967 \AtBeginDocument{% in case packages change definition
968 \let\LWR@orig@trivlist\@trivlist
969 \let\LWR@origtrivlist\trivlist
970 \let\LWR@origendtrivlist\endtrivlist
971 \LetLtxMacro\LWR@origitem\item
972 \LetLtxMacro\LWR@origitemize\itemize
973 \LetLtxMacro\LWR@endorigitemize\enditemize
974 \LetLtxMacro\LWR@origenumerate\enumerate
975 \LetLtxMacro\LWR@endorigenumerate\endenumerate
976 \LetLtxMacro\LWR@origdescription\description
977 \LetLtxMacro\LWR@endorigdescription\enddescription
978 \let\LWR@orig@mklab\@mklab
979 \let\LWR@origmakelabel\makelabel
980 \let\LWR@orig@donoparitem\@donoparitem
981 \LetLtxMacro\LWR@orig@item\@item
982 \let\LWR@orig@nbitem\@nbitem
983 }
984
985 \let\LWR@origpar\par
986
987 \LetLtxMacro\LWR@origfootnote\footnote
988 \let\LWR@orig@mpfootnotetext\@mpfootnotetext
```

```

989
990 \let\LWR@origclearpage\clearpage
991
992
993 \AtBeginDocument{% in case packages change definition
994 \LetLtxMacro\LWR@origline\hline%
995 \LetLtxMacro\LWR@origcline\cline%
996 }

997 \end{warpHTML}

```

36 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: 998 \begin{warpHTML}

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

```
999 \AtBeginDocument{
```

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

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

```
1000 \LetLtxMacro\LWR@origt\t
```

For \pdfTeX , additional work is required:

```

1001 \ifPDFTeX
1002 \LetLtxMacro\LWR@origequalaccent\=
1003 \LetLtxMacro\LWR@origdotaccent\..
1004 \LetLtxMacro\LWR@origu\u
1005 \LetLtxMacro\LWR@origv\v
1006 \LetLtxMacro\LWR@origc\c
1007 \LetLtxMacro\LWR@origd\d
1008 \LetLtxMacro\LWR@origb\b

```

The HTML redefinitions follow.

For \pdfTeX , Unicode diacritical marks are used:

```

1009 \renewcommand*{\=}[1]{\#1\HTMLUnicode{0305}}
1010 \renewcommand*{\..}[1]{\#1\HTMLUnicode{0307}}

```

```

1011 \renewcommand*{\u}[1]{#1\HTMLUnicode{0306}}
1012 \renewcommand*{\v}[1]{#1\HTMLUnicode{030C}}
1013 \renewcommand*{\c}[1]{#1\HTMLUnicode{0327}}
1014 \renewcommand*{\d}[1]{#1\HTMLUnicode{0323}}
1015 \renewcommand*{\b}[1]{#1\HTMLUnicode{0331}}
1016 \fi

```

For all engines, a Unicode diacritical tie is used:

```

1017 \def\LWR@t#1#2{#1\HTMLUnicode{0361}#2}
1018 \renewcommand*{\t}[1]{\LWR@t#1}

```

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

```

1019 \ifPDFTeX
1020 \newcommand*{\LWR@restoreorigaccents}{%
1021 \LetLtxMacro\=\LWR@origequalaccent%
1022 \LetLtxMacro\.\LWR@origdotaccent%
1023 \LetLtxMacro\u\LWR@origu%
1024 \LetLtxMacro\v\LWR@origv%
1025 \LetLtxMacro\t\LWR@origt%
1026 \LetLtxMacro\c\LWR@origc%
1027 \LetLtxMacro\d\LWR@origd%
1028 \LetLtxMacro\b\LWR@origb%
1029 }%
1030 \else% XeLaTeX, LuaLaTeX:
1031 \newcommand*{\LWR@restoreorigaccents}{%
1032 \LetLtxMacro\t\LWR@origt%
1033 }%
1034 \fi%
1035 }% AtBeginDocument

1036 \end{warpHTML}

```

37 Configuration Files

```

1037 \begin{warpprint}
1038 \typeout{lwarp: generating configuration files}
1039 \end{warpprint}

```

37.1 project_html.tex

File `project_html.tex` Used to allow an HTML version of the document to exist alongside the print version.

Only write `\jobname_html.tex` if generating the print version.

```

1040 \begin{warpprint}
1041 \immediate\openout\LWR@quickfile=\jobname_html.tex
1042 \immediate\write\LWR@quickfile{%
1043 \detokenize{\PassOptionsToPackage}%
1044 {warpHTML,BaseJobname=\jobname}{lwarp}%
1045 }
1046 \immediate\write\LWR@quickfile{%
1047 \detokenize{\input}\string{\jobname.tex\string }%
1048 }
1049 \immediate\closeout\LWR@quickfile
1050 \end{warpprint}

```

37.2 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:

```

1051 \begin{warpprint}
1052 \ifcsdef\LWR@quickfile{}\{\newwrite\LWR@quickfile}\}
1053 \immediate\openout\LWR@quickfile=lwarpmk.conf
1054 \ifbool{usingOSWindows}{
1055 \immediate\write\LWR@quickfile{opsystem = "Windows"}
1056 }{
1057 \immediate\write\LWR@quickfile{opsystem = "Unix"}
1058 }
1059 \ifPDFTeX
1060 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
1061 \fi
1062 \ifXeTeX
1063 \immediate\write\LWR@quickfile{latexname = "xelatex"}
1064 \fi
1065 \ifLuaTeX
1066 \immediate\write\LWR@quickfile{latexname = "lualatex"}

```

```

1067 \fi
1068 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
1069 \immediate\write\LWR@quickfile{%
1070 homehtmlfilename = "\HomeHTMLFilename"%
1071 }
1072 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
1073 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}}
1074 \immediate\write\LWR@quickfile{shellescape = "\ifshellescape true\else false\fi"}
1075 \immediate\write\LWR@quickfile{xindylanguage = "\LWR@xindyLanguage"}
1076 \immediate\write\LWR@quickfile{xindycodepage = "\LWR@xindyCodepage"}
1077 \immediate\write\LWR@quickfile{xindystyle = "\LWR@xindyStyle"}
1078 \immediate\write\LWR@quickfile{pdftotextenc = "\LWR@pdftotextEnc"}
1079 \immediate\closeout\LWR@quickfile
1080 \end{warpprint}

```

37.3 project.lwarpmkconf

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

```

1081 \begin{warpprint}
1082 \ifcsdef{LWR@quickfile}{\newwrite{LWR@quickfile}}
1083 \immediate\openout\LWR@quickfile=\jobname.lwarpmkconf
1084 \ifbool{usingOSWindows}{
1085 \immediate\write\LWR@quickfile{opsystem = "Windows"}
1086 }{
1087 \immediate\write\LWR@quickfile{opsystem = "Unix"}
1088 }
1089 \ifPDFTeX
1090 \immediate\write\LWR@quickfile{latexname = "pdflatex"}
1091 \fi
1092 \ifXeTeX
1093 \immediate\write\LWR@quickfile{latexname = "xelatex"}
1094 \fi
1095 \ifLuaTeX
1096 \immediate\write\LWR@quickfile{latexname = "lualatex"}
1097 \fi
1098 \immediate\write\LWR@quickfile{sourcename = "\jobname"}
1099 \immediate\write\LWR@quickfile{%
1100 homehtmlfilename = "\HomeHTMLFilename"%
1101 }
1102 \immediate\write\LWR@quickfile{htmlfilename = "\HTMLFilename"}
1103 \immediate\write\LWR@quickfile{latexmk = "\ifbool{LWR@latexmk}{true}{false}}
1104 \immediate\write\LWR@quickfile{shellescape = "\ifshellescape true\else false\fi"}
1105 \immediate\write\LWR@quickfile{xindylanguage = "\LWR@xindyLanguage"}
1106 \immediate\write\LWR@quickfile{xindycodepage = "\LWR@xindyCodepage"}
1107 \immediate\write\LWR@quickfile{xindystyle = "\LWR@xindyStyle"}
1108 \immediate\write\LWR@quickfile{pdftotextenc = "\LWR@pdftotextEnc"}

```

```

1109 \immediate\closeout\LWR@quickfile
1110 \end{warpprint}

```

37.4 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.

```

1111 \begin{warpprint}
1112 \begin{filecontents*}{lwarp.css}
1113 /*
1114  CSS stylesheet for the LaTeX lwarp package
1115  Copyright 2016-2018 Brian Dunn -- BD Tech Concepts LLC
1116 */
1117
1118
1119 /* a fix for older browsers: */
1120 header, section, footer, aside, nav, main,
1121     article, figure { display: block; }
1122
1123
1124 A:link {color:#000080 ; text-decoration: none ; }
1125 A:visited {color:#800000 ; }
1126 A:hover {color:#000080 ; text-decoration: underline ;}
1127 A:active {color:#800000 ; }
1128
1129 a.tocpart {display: inline-block ; margin-left: 0em ;
1130     font-weight: bold ;}
1131 a.tocchapter {display: inline-block ; margin-left: 0em ;
1132     font-weight: bold ;}
1133 a.tocsection {display: inline-block ; margin-left: 1em ;
1134     text-indent: -.5em ; font-weight: bold ; }
1135 a.tocsubsection {display: inline-block ; margin-left: 2em ;
1136     text-indent: -.5em ; }
1137 a.tocsubsubsection {display: inline-block ; margin-left: 3em ;
1138     text-indent: -.5em ; }
1139 a.tocparagraph {display: inline-block ; margin-left: 4em ;
1140     text-indent: -.5em ; }
1141 a.tocsubparagraph {display: inline-block ; margin-left: 5em ;
1142     text-indent: -.5em ; }
1143 a.tocfigure {margin-left: 0em}
1144 a.tocsubfigure {margin-left: 2em}
1145 a.toctable {margin-left: 0em}
1146 a.tocsubtable {margin-left: 2em}
1147 a.toctheorem {margin-left: 0em}

```

```
1148 a.toclstlisting {margin-left: 0em}
1149
1150 body {
1151     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1152         "Lucida Bright", Georgia, serif;
1153     background: #FAF7F4 ;
1154     color: black ;
1155     margin: 0em ;
1156     padding: 0em ;
1157     font-size: 100% ;
1158     line-height: 1.2 ;
1159 }
1160
1161 p {margin: 1.5ex 0em 1.5ex 0em ;}
1162 table p {margin: .5ex 0em .5ex 0em ;}
1163
1164 /* Holds a section number to add space between it and the name */
1165 span.sectionnumber { margin-right: 0em }
1166
1167 /* Inserted in front of index lines */
1168 span.indexitem {margin-left: 0em}
1169 span.indexsubitem {margin-left: 2em}
1170 span.indexsubsubitem {margin-left: 4em}
1171
1172 div.hidden, span.hidden { display: none ; }
1173
1174 kbd, span.texttt {
1175     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1176         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1177         "Courier New", monospace;
1178     font-size: 100% ;
1179 }
1180
1181 pre { padding: 3pt ; }
1182
1183 span.strong, span.textbf, div.strong, div.textbf { font-weight: bold; }
1184
1185 span.textit, div.textit { font-style: italic; }
1186
1187 span.textmd, div.textmd { font-weight: normal; }
1188
1189 span.textsc, div.textsc { font-variant: small-caps; }
1190
1191 span.textsl, div.textsl { font-style: oblique; }
1192
1193 span.textup, div.textup { font-variant: normal; }
1194
1195 span.textrm, div.textrm {
1196     font-family: "DejaVu Serif", "Bitstream Vera Serif",
1197         "Lucida Bright", Georgia, serif;
```

```
1198 }
1199
1200 span.textsf, div.textsf {
1201     font-family: "DejaVu Sans", "Bitstream Vera Sans",
1202         Geneva, Verdana, sans-serif ;
1203 }
1204
1205 span.textcircled { border: 1px solid black ; border-radius: 1ex ; }
1206
1207 span.underline {
1208     text-decoration: underline ;
1209     text-decoration-skip: auto ;
1210 }
1211
1212 span.overline {
1213     text-decoration: overline ;
1214     text-decoration-skip: auto ;
1215 }
1216
1217 /* for diagbox */
1218 div.diagboxtitleN { border-bottom: 1px solid gray }
1219 div.diagboxtitleS { border-top: 1px solid gray }
1220
1221 div.diagboxE {
1222     padding-left: 2em ;
1223     text-align: right ;
1224 }
1225
1226 div.diagboxW {
1227     padding-right: 2em ;
1228     text-align: left ;
1229 }
1230
1231
1232
1233 /* For realscripts */
1234 .supsubscript {
1235     display: inline-block;
1236     text-align:left ;
1237 }
1238
1239 .supsubscript sup,
1240 .supsubscript sub {
1241     position: relative;
1242     display: block;
1243     font-size: .5em;
1244     line-height: 1;
1245 }
1246
1247 .supsubscript sup {
```

```
1248     top: .5em;
1249 }
1250
1251 .supsubscript sub {
1252     top: .5em;
1253 }
1254
1255 div.attribution p {
1256     text-align: right ;
1257     font-size: 80%
1258 }
1259
1260 span.poemtitle {
1261     font-size: 120% ; font-weight: bold;
1262 }
1263
1264 pre.tabbing {
1265     font-family: "Linux Libertine Mono O", "Lucida Console",
1266                 "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1267                 "Liberation Mono", "FreeMono", "Andale Mono",
1268                 "Nimbus Mono L", "Courier New", monospace;
1269 }
1270
1271 blockquote {
1272     margin-left: 0px ;
1273     margin-right: 0px ;
1274 }
1275
1276 /* quotchap is for the quotchap package */
1277 div.quotchap {
1278     font-style: oblique ;
1279     overflow-x: auto ;
1280     margin-left: 2em ;
1281     margin-right: 2em ;
1282 }
1283
1284 blockquote p, div.quotchap p {
1285     line-height: 1.5;
1286     text-align: left ;
1287     font-size: .85em ;
1288     margin-left: 3em ;
1289     margin-right: 3em ;
1290 }
1291
1292 /* qauthor is for the quotchap package */
1293 div.qauthor {
1294     display: block ;
1295     text-align: right ;
1296     margin-left: auto ;
1297     margin-right: 2em ;
```

```
1298 font-size: 80% ;
1299 font-variant: small-caps;
1300 }
1301
1302 div.qauthor p {
1303   text-align: right ;
1304 }
1305
1306 blockquotation {
1307   margin-left: 0px ;
1308   margin-right: 0px ;
1309 }
1310
1311 blockquotation p {
1312   line-height: 1.5;
1313   text-align: left ;
1314   font-size: .85em ;
1315   margin-left: 3em ;
1316   margin-right: 3em ;
1317 }
1318
1319 div.epigraph, div.dictum {
1320   line-height: 1.2;
1321   text-align: left ;
1322   padding: 3ex 1em 0ex 1em ;
1323   /* margin: 3ex auto 3ex auto ; */ /* Epigraph centered */
1324   margin: 3ex 1em 3ex auto ; /* Epigraph to the right */
1325   /* margin: 3ex 1em 3ex 1em ; */ /* Epigraph to the left */
1326   font-size: .85em ;
1327   max-width: 27em ;
1328 }
1329
1330
1331
1332 div.epigraphsource, div.dictumauthor {
1333   text-align:right ;
1334   margin-left:auto ;
1335   /* max-width: 50% ; */
1336   border-top: 1px solid #A0A0A0 ;
1337   padding-bottom: 3ex ;
1338   line-height: 1.2;
1339 }
1340
1341 div.epigraph p, div.dictum p { padding: .5ex ; margin: 0ex ;}
1342 div.epigraphsource p, div.dictumauthor p { padding: .5ex 0ex 0ex 0ex ; margin: 0ex ;}
1343 div.dictumauthor { font-style:italic }
1344
1345
1346 /* lettrine package: */
1347 span.lettrine { font-size: 4ex ; float: left ; }
```

```
1348 span.lettrinetext { font-variant: small-caps ; }
1349
1350 /* ulem and soul packages: */
1351 span.uline {
1352     text-decoration: underline ;
1353     text-decoration-skip: auto ;
1354 }
1355
1356 span.uuline {
1357     text-decoration: underline ;
1358     text-decoration-skip: auto ;
1359     text-decoration-style: double ;
1360 }
1361
1362 span.uwave {
1363     text-decoration: underline ;
1364     text-decoration-skip: auto ;
1365     text-decoration-style: wavy ;
1366 }
1367
1368 span.sout {
1369     text-decoration: line-through ;
1370 }
1371
1372 span.xout {
1373     text-decoration: line-through ;
1374 }
1375
1376 span.dashuline {
1377     text-decoration: underline ;
1378     text-decoration-skip: auto ;
1379     text-decoration-style: dashed ;
1380 }
1381
1382 span.dotuline {
1383     text-decoration: underline ;
1384     text-decoration-skip: auto ;
1385     text-decoration-style: dotted ;
1386 }
1387
1388 span.letterspacing { letter-spacing: .2ex ; }
1389
1390 span.capsspacing {
1391     font-variant: small-caps ;
1392     letter-spacing: .1ex ;
1393 }
1394
1395 span.highlight { background: #F8E800 ; }
1396
1397
```

```
1398
1399
1400html body {
1401  margin: 0 ;
1402  line-height: 1.2;
1403}
1404
1405
1406body div {
1407  margin: 0ex;
1408}
1409
1410
1411h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
1412{
1413  font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
1414             "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
1415             "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
1416             "Times New Roman", serif;
1417  font-style: normal ;
1418  font-weight: bold ;
1419  text-align: left ;
1420}
1421
1422h1 { /* title of the entire website, used on each page */
1423  text-align: center ;
1424  font-size: 2.5em ;
1425  padding: .4ex 0em 0ex 0em ;
1426}
1427h2 { font-size: 2.25em }
1428h3 { font-size: 2em }
1429h4 { font-size: 1.75em }
1430h5 { font-size: 1.5em }
1431h6 { font-size: 1.25em }
1432span.paragraph {font-size: 1em ; font-variant: normal ;
1433  margin-right: 1em ; }
1434span.subparagraph {font-size: 1em ; font-variant: normal ;
1435  margin-right: 1em ; }
1436
1437div.minisec {
1438  font-family: "DejaVu Sans", "Bitstream Vera Sans",
1439             Geneva, Verdana, sans-serif ;
1440  font-style: normal ;
1441  font-weight: bold ;
1442  text-align: left ;
1443}
1444
1445/* Title of the file */
1446h1 {
1447  margin: 0ex 0em 0ex 0em ;
```

```
1448 line-height: 1.3;
1449 text-align: center ;
1450 }
1451
1452 /* Part */
1453 h2 {
1454     margin: 1ex 0em 1ex 0em ;
1455     line-height: 1.3;
1456     text-align: center ;
1457 }
1458
1459 /* Chapter */
1460 h3 {
1461     margin: 3ex 0em 1ex 0em ;
1462     line-height: 1.3;
1463 }
1464
1465 /* Section */
1466 h4 {
1467     margin: 3ex 0em 1ex 0em ;
1468     line-height: 1.3;
1469 }
1470
1471 /* Sub-Section */
1472 h5 {
1473     margin: 3ex 0em 1ex 0em ;
1474     line-height: 1.3;
1475 }
1476
1477 /* Sub-Sub-Section */
1478 h6 {
1479     margin: 3ex 0em 1ex 0em ;
1480     line-height: 1.3;
1481 }
1482
1483
1484 div.titlepage {
1485     text-align: center ;
1486 }
1487
1488 .footnotes {
1489     font-size: .85em ;
1490     margin: 3ex 2em 0ex 2em ;
1491     border-top: 1px solid silver ;
1492 }
1493
1494 .marginpar, .marginparblock {
1495     max-width: 50%;
1496     float: right;
1497     text-align: left;
```

```
1498     margin: 1ex 0.5em 1ex 1em ;
1499     padding: 1ex 0.5em 1ex 0.5em ;
1500     font-size: 85% ;
1501     border-top: 1px solid silver ;
1502     border-bottom: 1px solid silver ;
1503     overflow-x: auto;
1504 }
1505
1506 .marginpar br { margin-bottom: 2ex ; }
1507
1508 div.marginblock, div.marginparblock {
1509     max-width:50%;
1510     float:right;
1511     text-align:left;
1512     margin: 1ex 0.5em 1ex 1em ;
1513     padding: 1ex 0.5em 1ex 0.5em ;
1514     overflow-x: auto;
1515 }
1516
1517 div.marginblock div.minipage,
1518 div.marginparblock div.minipage {
1519     display: block ;
1520     margin: 0pt auto 0pt auto ;
1521 }
1522
1523 div.marginblock div.minipage p ,
1524 div.marginparblock div.minipage p
1525     { font-size: 85%}
1526
1527 div.marginblock br ,
1528 div.marginparblock br
1529     { margin-bottom: 2ex ; }
1530
1531
1532 section.textbody div.footnotes{
1533     margin: 3ex 2em 0ex 2em ;
1534     border-bottom: 2px solid silver ;
1535 }
1536
1537 .footnoteheader {
1538     border-top: 2px solid silver ;
1539     margin-top: 3ex ;
1540     padding-top: 1ex ;
1541     font-weight: bold ;
1542 }
1543
1544 .mpfootnotes {
1545     text-align: left ;
1546     font-size: .85em ;
1547     margin-left: 1em ;
```

```
1548     border-top: 1px solid silver ;
1549 }
1550
1551 /* Remove footnote top border in the title page. */
1552 div.titlepage div.mpfootnotes {
1553     border-top: none ;
1554 }
1555
1556
1557
1558 ol {
1559     margin: 1ex 1em 1ex 0em;
1560     line-height: 1.2;
1561 }
1562
1563 ul, body dir, body menu {
1564     margin: 3ex 1em 3ex 0em;
1565     line-height: 1.2;
1566 }
1567
1568 li { margin: 0ex 0em 1ex 0em; }
1569
1570 html {
1571     margin: 0;
1572     padding: 0;
1573 }
1574
1575 .programlisting {
1576     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1577         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1578         "Courier New", monospace;
1579     margin: 1ex 0ex 1ex 0ex ;
1580     padding: .5ex 0pt .5ex 0pt ;
1581     overflow-x: auto;
1582 }
1583
1584 section.textbody>pre.programlisting {
1585 border-top: 1px solid silver ;
1586 border-bottom: 1px solid silver ;
1587 }
1588
1589
1590 div.displaymath {
1591     text-align: center ;
1592 }
1593
1594 div.displaymathnumbered {
1595     text-align: right ;
1596     margin-left: 5% ;
1597     margin-right: 5% ;
```

```
1598     min-width: 2.5in ;
1599 }
1600
1601 @media all and (min-width: 400px) {
1602     div.displaymathnumbered {
1603         margin-left: 10% ;
1604         margin-right: 10% ;
1605     }
1606 }
1607
1608 @media all and (min-width: 800px) {
1609     div.displaymathnumbered {
1610         margin-right: 20% ;
1611     }
1612 }
1613
1614 @media all and (min-width: 1200px) {
1615     div.displaymathnumbered {
1616         margin-right: 30% ;
1617     }
1618 }
1619
1620
1621 .inlineprogramlisting {
1622     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
1623                 "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
1624                 "Courier New", monospace;
1625     overflow-x: auto;
1626 }
1627
1628 span.listinglabel {
1629     display: inline-block ;
1630     font-size: 70% ;
1631     width: 4em ;
1632     text-align: right ;
1633     margin-right: 2em ;
1634 }
1635
1636 div.abstract {
1637     margin: 2em 5% 2em 5% ;
1638     padding: 1ex 1em 1ex 1em ;
1639     /* font-weight: bold ; */
1640     font-size: 90% ;
1641     text-align: left ;
1642 }
1643
1644 div.abstract dl {line-height:1.5;}
1645 div.abstract dt {color:#304070;}
1646
1647 div.abstracttitle{
```

```
1648     font-family: "URW Classico", Optima, "Linux Biolinum O",
1649         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1650         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1651     font-weight:bold;
1652     font-size:1.25em;
1653     text-align: center ;
1654 }
1655
1656 span.abstractrunintitle{
1657     font-family: "URW Classico", Optima, "Linux Biolinum O",
1658         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
1659         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
1660     font-weight:bold;
1661 }
1662
1663
1664 .verbatim {
1665     overflow-x: auto ;
1666 }
1667
1668 .alltt {
1669     overflow-x: auto ;
1670 }
1671
1672
1673 .bverbatim {
1674     margin: 1ex 0pt 1ex 0pt ;
1675     padding: .5ex 0pt .5ex 0pt ;
1676     overflow-x: auto ;
1677 }
1678
1679 .lverbatim {
1680     margin: 1ex 0pt 1ex 0pt ;
1681     padding: .5ex 0pt .5ex 0pt ;
1682     overflow-x: auto ;
1683 }
1684
1685 .fancyvrb {
1686     font-size:.85em ;
1687     margin: 3ex 0pt 3ex 0pt
1688 }
1689
1690 .fancyvrblabel {
1691     font-weight:bold;
1692     text-align: center ;
1693 }
1694
1695
1696 .verse {
1697     font-family: "Linux Libertine Mono O", "Lucida Console",
```

```
1698     "Droid Sans Mono", "DejaVu Mono", "Bitstream Vera Mono",
1699     "Liberation Mono", "FreeMono", "Andale Mono",
1700     "Nimbus Mono L", "Courier New", monospace;
1701     margin-left: 1em ;
1702 }
1703
1704
1705 div.singlespace { line-height: 1.2 ; }
1706 div.onehalfspace { line-height: 1.5 ; }
1707 div.doublespace { line-height: 2 ; }
1708
1709
1710 /* Word processor format output: */
1711 div.wpfigure { border: 1px solid red ; margin: .5ex ; padding: .5ex ; }
1712 div.wptable { border: 1px solid blue ; margin: .5ex ; padding: .5ex ; }
1713 div.wpminipage { border: 1px solid green ; margin: .5ex ; padding: .5ex ;}
1714
1715
1716
1717
1718 /* Minipage environments, vertically aligned to top, center, bottom: */
1719 .minipage, .fminipage, .fcolorminipage {
1720     /* display: inline-block ; */
1721     /* Mini pages which follow each other will be tiled. */
1722     margin: .25em .25em .25em .25em;
1723     padding: .25em .25em .25em .25em;
1724     display: inline-flex;
1725     flex-direction: column ;
1726     overflow: auto;
1727 }
1728
1729 /* Paragraphs in the flexbox did not collapse their margins. */
1730 /* Have not yet researched this. */
1731 .minipage p {margin: .75ex 0em .75ex 0em ;}
1732
1733 .fbboxBlock .minipage, .colorbox .minipage, .colorboxBlock .minipage,
1734 .fcolorbox .minipage, .fcolorboxBlock .minipage
1735     {border: none ; background: none;}
1736
1737 .fbbox, .fbboxBlock { border: 1px solid black ; }
1738
1739 .fbbox, .fbboxBlock, .fcolorbox, .fcolorboxBlock, .colorbox, .colorboxBlock,
1740 .fminipage, .fcolorminipage
1741     {display: inline-block}
1742
1743 .shadowbox, .shabox {
1744     border: 1px solid black;
1745     box-shadow: 3px 3px 3px #808080 ;
1746     border-radius: 0px ;
1747     padding: .4ex .3em .4ex .3em ;
```

```
1748     margin: 0pt .3ex 0pt .3ex ;
1749     display: inline-block ;
1750 }
1751
1752 .doublebox {
1753     border: 3px double black;
1754     border-radius: 0px ;
1755     padding: .4ex .3em .4ex .3em ;
1756     margin: 0pt .3ex 0pt .3ex ;
1757     display: inline-block ;
1758 }
1759
1760 .ovalbox, .Ovalbox {
1761     border: 1px solid black;
1762     border-radius: 1ex ;
1763     padding: .4ex .3em .4ex .3em ;
1764     margin: 0pt .3ex 0pt .3ex ;
1765     display: inline-block ;
1766 }
1767
1768 .Ovalbox { border-width: 2px ; }
1769
1770 .framebox {
1771     border: 1px solid black;
1772     border-radius: 0px ;
1773     padding: .3ex .2em 0ex .2em ;
1774     margin: 0pt .1ex 0pt .1ex ;
1775     display: inline-block ;
1776 }
1777
1778
1779 .mdframed {
1780 /*     padding: 0ex ; */
1781 /*     border: 1px solid black; */
1782 /*     border-radius: 0px ; */
1783     padding: 0ex ;
1784     margin: 3ex 5% 3ex 5% ;
1785 /*     display: inline-block ; */
1786 }
1787
1788 .mdframed p { padding: 0ex .5em 0ex .5em ; }
1789
1790 .mdframed dl { padding: 0ex .5em 0ex .5em ; }
1791
1792 .mdframedtitle {
1793     padding: .5em ;
1794     display: block ;
1795     font-size: 130% ;
1796     margin-bottom: 1ex ;
1797 }
```

```
1798
1799 .mdframedsubtitle {
1800     padding: 0ex .5em 0ex .5em ;
1801     display: block ;
1802     font-size: 115% ;
1803 }
1804
1805 .mdframedsubsubtitle {
1806     padding: 0ex .5em 0ex .5em ;
1807     display: block ;
1808 }
1809
1810 .mdtheorem {
1811     padding: 0ex .5em 0ex .5em ;
1812     margin: 3ex 5% 3ex 5% ;
1813 /*    display: inline-block ; */
1814 }
1815
1816
1817 /* framed package */
1818 .framed, pre.boxedverbatim, fcolorbox {
1819     margin: 3ex 0em 3ex 0em ;
1820     border: 1px solid black;
1821     border-radius: 0px ;
1822     padding: .3ex 1em 0ex 1em ;
1823     display: block ;
1824 }
1825
1826 .shaded {
1827     margin: 3ex 0em 3ex 0em ;
1828     padding: .3ex 1em .3ex 1em ;
1829     display: block ;
1830 }
1831
1832 .snugframed {
1833     margin: 3ex 0em 3ex 0em ;
1834     border: 1px solid black;
1835     border-radius: 0px ;
1836     display: block ;
1837 }
1838
1839 .framedleftbar {
1840     margin: 3ex 0em 3ex 0em ;
1841     border-left: 3pt solid black;
1842     border-radius: 0px ;
1843     padding: .3ex .2em .3ex 1em ;
1844     display: block ;
1845 }
1846
1847 .framedtitle {
```

```
1848     margin: 0em ;
1849     padding: 0em ;
1850     font-size: 130%
1851 }
1852
1853 .framedtitle p { padding: .3em }
1854
1855
1856
1857 dl {
1858     margin: 1ex 2em 1ex 0em;
1859     line-height: 1.3;
1860 }
1861
1862 dl dt {
1863     margin-top: 1ex;
1864     margin-left: 1em ;
1865     font-weight: bold;
1866 }
1867
1868 dl dd p { margin-top: 0em; }
1869
1870
1871 nav {
1872     font-family: "URW Classico", Optima, "Linux Biolinum O",
1873         "DejaVu Sans", "Bitstream Vera Sans",
1874         Geneva, Verdana, sans-serif ;
1875     margin-bottom: 4ex ;
1876 }
1877
1878 nav p {
1879     line-height: 1.2 ;
1880     margin-top:.5ex ;
1881     margin-bottom:.5ex;
1882     font-size: .9em ;
1883 }
1884
1885
1886
1887 img, img.hyperimage, img.borderimage {
1888     max-width: 600px;
1889     border: 1px solid silver;
1890     box-shadow: 3px 3px 3px #808080 ;
1891     padding: .5% ;
1892     margin: .5% ;
1893     background: none ;
1894 }
1895
1896 img.inlineimage{
1897     padding: 0px ;
```

```
1898     box-shadow: none ;
1899     border: none ;
1900     background: none ;
1901     margin: 0px ;
1902     display: inline-block ;
1903     border-radius: 0px ;
1904 }
1905
1906 img.logoimage{
1907     max-width: 300px ;
1908     box-shadow: 3px 3px 3px #808080 ;
1909     border: 1px solid black ;
1910     background:none ;
1911     padding:0 ;
1912     margin:.5ex ;
1913     border-radius: 10px ;
1914 }
1915
1916
1917 .section {
1918 /*
1919     To have each section float relative to each other:
1920 */
1921 /*
1922     display: block ;
1923     float: left ;
1924     position: relative ;
1925     background: white ;
1926     border: 1px solid silver ;
1927     padding: .5em ;
1928 */
1929     margin: 0ex .5em 0ex .5em ;
1930     padding: 0 ;
1931 }
1932
1933
1934 figure {
1935     margin: 5ex auto 5ex auto ;
1936     padding: 1ex 1em 1ex 1em ;
1937     overflow-x: auto ;
1938 }
1939
1940
1941 /* To automatically center images in figures: */
1942 /*
1943 figure img.inlineimage {
1944     margin: 0ex auto 0ex auto ;
1945     display: block ;
1946 }
1947 */
```

```
1948
1949 /* To automatically center minipages in figures: */
1950 /*
1951 figure div.minipage, figure div.minipage div.minipage {
1952     margin: 1ex auto 1ex auto ;
1953     display: block ;
1954 }
1955 */
1956
1957 figure div.minipage p { font-size: 85% ; }
1958
1959 figure.subfigure, figure.subtable {
1960     display: inline-block ; margin: 3ex 1em 3ex 1em ;
1961 }
1962
1963 figcaption .minipage { margin:0 ; padding: 0 }
1964
1965 div.minipage figure { border: none ; box-shadow: none ; }
1966
1967 div.floatrow { text-align: center; }
1968
1969 div.floatrow figure { display: inline-block ; margin: 1ex 2% ; }
1970
1971 div.floatfoot { font-size: .85em ;
1972     border-top: 1px solid silver ; line-height: 1.2 ; }
1973
1974 figcaption , .lstlistingtitle {
1975     font-size: .85em ;
1976     text-align: center ;
1977     font-weight: bold ;
1978     margin-top: 1ex ;
1979     margin-bottom: 1ex ;
1980 }
1981
1982 figure.subfigure figcaption, figure.subtable figcaption {
1983     border-bottom: none ; background: none ;
1984 }
1985
1986 div.nonfloatcaption {
1987     margin: 1ex auto 1ex auto ;
1988     font-size: .85em ;
1989     text-align: center ;
1990     font-weight: bold ;
1991 }
1992
1993 /* For a \RawCaption inside a minipage inside a figure's floatrow: */
1994 figure div.floatrow div.minipage figcaption {
1995     border: none ;
1996     background: none ;
1997 }
```

```

1998
1999
2000 /* For packages such as float, rotfloat, and algorithm2e: */
2001
2002 figure.boxed, figure.boxruled {
2003     border: 1px solid black ;
2004 }
2005
2006 figure.ruled {
2007     border-top: 1px solid black ;
2008     border-bottom: 1px solid black ;
2009     border-left: 0px ;
2010     border-right: 0px ;
2011     border-radius: 0px ;
2012     background: none ;
2013     box-shadow: none ;
2014 }
2015
2016 figure.ruled figcaption, figure.boxruled figcaption {
2017     border-top: 1px solid silver ;
2018     border-bottom: 1px solid silver ;
2019 }
2020
2021
2022 table {
2023     margin: 1ex auto 1ex auto ;
2024     border-collapse: separate ;
2025     border-spacing: 0px ;
2026     line-height: 1.3 ;
2027 }
2028
2029 tr.hline td {border-top: 1px solid #808080 ; margin-top: 0ex ;
2030     margin-bottom: 0ex ; } /* for \hline */
2031
2032 tr.tbrule td {border-top: 1px solid black ; margin-top: 0ex ;
2033     margin-bottom: 0ex ; } /* for \toprule, \bottomrule */
2034
2035 td {padding: .5ex .5em .5ex .5em ;}
2036
2037 table td.tdl { text-align: left ; vertical-align: middle ; }
2038 table td.tdc { text-align: center ; vertical-align: middle ; }
2039 table td.tdat { text-align: center ; vertical-align: middle ; padding: 0px ; margin: 0px ; }
2040 table td.tdbang { text-align: center ; vertical-align: middle ; }
2041 table td.tdr { text-align: right ; vertical-align: middle ; }
2042 table td.tdp { text-align: left ; vertical-align: bottom ; }
2043 table td.tdm { text-align: left ; vertical-align: middle ; }
2044 table td.tdb { text-align: left ; vertical-align: top ; }
2045 table td.tdP { text-align: center ; vertical-align: bottom ; }
2046 table td.tdM { text-align: center ; vertical-align: middle ; }
2047 table td.tdB { text-align: center ; vertical-align: top ; }

```

```

2048
2049 table td.tvertbarl { border-left: 1px solid black }
2050 table td.tvertbarldouble { border-left: 4px double black }
2051 table td.tvertbarr { border-right: 1px solid black }
2052 table td.tvertbarrdouble { border-right: 4px double black }
2053
2054 table td.tvertbarldash { border-left: 1px dashed black }
2055 table td.tvertbarldoubledash { border-left: 2px dashed black }
2056 table td.tvertbarrdash { border-right: 1px dashed black }
2057 table td.tvertbarrdoubledash { border-right: 2px dashed black }
2058
2059
2060 /* for cmidrules: */
2061 table td.tdrule {
2062     border-top: 1px solid #A0A0A0 ;
2063 }
2064
2065 table td.tdrulel {
2066     border-top-left-radius:.5em ;
2067     border-top: 1px solid #A0A0A0 ;
2068 }
2069
2070 table td.tdruler {
2071     border-top-right-radius:.5em ;
2072     border-top: 1px solid #A0A0A0 ;
2073 }
2074
2075 table td.tdrulelr {
2076     border-top-left-radius:.5em ;
2077     border-top-right-radius:.5em ;
2078     border-top: 1px solid #A0A0A0 ;
2079 }
2080
2081
2082 /* Margins of paragraphs inside table cells: */
2083 td.tdp p , td.tdprule p , td.tdP p , td.tdPrule p { padding-top: 1ex ;
2084     padding-bottom: 1ex ; margin: 0ex ; }
2085 td.tdm p , td.tmbrule p , td.tdM p , td.tdMrule p { padding-top: 1ex ;
2086     padding-bottom: 1ex ; margin: 0ex ; }
2087 td.tdb p , td.tdbrule p , td.tdB p , td.tdBrule p { padding-top: 1ex ;
2088     padding-bottom: 1ex ; margin: 0ex ; }
2089
2090 td.tdp , td.tdprule , td.tdP , td.tdPrule
2091     { padding: 0ex .5em 0ex .5em ; }
2092 td.tdm , td.tdmrule , td.tdM , td.tdMrule
2093     { padding: 0ex .5em 0ex .5em ; }
2094 td.tdb , td.tdbrule , td.tdB , td.tdBrule
2095     { padding: 0ex .5em 0ex .5em ; }
2096
2097

```

```
2098 /* table notes: */
2099 .tnotes {
2100     margin: 0ex 5% 1ex 5% ;
2101     padding: 0.5ex 1em 0.5ex 1em;
2102     font-size:.85em;
2103     text-align: left ;
2104 }
2105
2106 .tnotes dl dt p {margin-bottom:0px;}
2107
2108 .tnoteitemheader {margin-right: 1em;}
2109
2110
2111 /* for colortbl and cell color */
2112 div.cellcolor {
2113     width: 100% ;
2114     padding: .5ex .5em .5ex .5em ;
2115     margin: -.5ex -.5em -.5ex -.5em ;
2116 }
2117
2118
2119 /* for bigdelim */
2120 .ldelim, .rdelim { font-size: 200% }
2121
2122
2123 /* center, flushleft, flushright environments */
2124 div.center{text-align:center;}
2125 div.center table {margin-left:auto;margin-right:auto;}
2126 div.flushleft{text-align:left;}
2127 div.flushleft table {margin-left:0em ; margin-right:auto;}
2128 div.flushright{text-align:right;}
2129 div.flushright table {margin-left:auto ; margin-right: 0em ;}
2130
2131
2132 /* Fancybox */
2133 div.Btrivlist table tr td {
2134     padding: .2ex 0em ;
2135 }
2136
2137
2138 /* program listing callouts: */
2139 span.callout {
2140     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2141         Geneva, Verdana, sans-serif ;
2142     border-radius: .5em;
2143     background-color:black;
2144     color:white;
2145     padding:0px .25em 0px .25em;
2146     margin: 0 ;
2147     font-weight: bold;
```

```
2148     font-size:.72em ;
2149 }
2150
2151 div.programlisting pre.verbatim span.callout{
2152     font-size: .85em ;
2153 }
2154
2155 span.verbatim {
2156     font-family: "DejaVu Mono", "Bitstream Vera Mono", "Lucida Console",
2157         "Nimbus Mono L", "Liberation Mono", "FreeMono", "Andale Mono",
2158         "Courier New", monospace;
2159 }
2160
2161
2162
2163 div.published
2164 {
2165     text-align: center ;
2166     font-variant: normal ;
2167     font-style: italic ;
2168     font-size: 1em ;
2169     margin: 3ex 0em 3ex 0em ;
2170 }
2171
2172 div.subtitle
2173 {
2174     text-align: center ;
2175     font-variant: normal ;
2176     font-style: italic ;
2177     font-size: 1.25em ;
2178     margin: 3ex 0em 3ex 0em ;
2179 }
2180
2181 div.subtitle p { margin: 1ex ; }
2182
2183 div.author
2184 {
2185     font-variant: normal ;
2186     font-style: normal ;
2187     font-size: 1em ;
2188     margin: 3ex 0em 3ex 0em ;
2189 }
2190
2191 div.oneauthor {
2192     display: inline-block ;
2193     margin: 3ex 1em 0ex 1em ;
2194 }
2195
2196 /*
2197 div.author table {
```

```
2198     margin: 3ex auto 0ex auto ;
2199     background: none ;
2200 }
2201
2202 div.author table tbody tr td { padding: .25ex ; }
2203 */
2204
2205 span.affiliation {font-size: .85em ; font-variant: small-caps; }
2206
2207 div.titledate {
2208     text-align: center ;
2209     font-size: .85em ;
2210     font-style: italic;
2211     margin: 6ex 0em 6ex 0em ;
2212 }
2213
2214
2215 nav.topnavigation{
2216     text-align: left ;
2217     padding: 0.5ex 1em 0.5ex 1em ;
2218 /*     margin: 2ex 0em 3ex 0em ; */
2219     margin: 0 ;
2220     border-bottom: 1px solid silver ;
2221     border-top: 1px solid silver ;
2222     clear:right ;
2223 }
2224
2225 nav.botnavigation{
2226     text-align: left ;
2227     padding: 0.5ex 1em 0.5ex 1em ;
2228 /*     margin: 3ex 0em 2ex 0em ; */
2229     margin: 0 ;
2230     border-top: 1px solid silver ;
2231     border-bottom: 1px solid silver ;
2232     clear:right ;
2233 }
2234
2235
2236 header{
2237     line-height: 1.2 ;
2238     font-size: 1em ;
2239 /*     border-bottom: 2px solid silver ; */
2240     margin: 0px ;
2241     padding: 0ex 1em 0ex 1em ;
2242     text-align:center ;
2243 }
2244
2245 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
2246
2247
```

```
2248 footer{
2249     font-size: .85em ;
2250     line-height: 1.2 ;
2251     margin-top: 1ex ;
2252     border-top: 2px solid silver ;
2253     padding: 2ex 1em 2ex 1em ;
2254     clear:right ;
2255     text-align:left ;
2256 }
2257
2258
2259 a.linkhome { font-weight:bold ; font-size: 1em ;}
2260
2261
2262 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
2263
2264 img.lateximage{
2265     padding: 0pt ;
2266     margin: 0pt ;
2267     box-shadow: none ;
2268     border: none ;
2269     background: none ;
2270     max-width: 100% ;
2271     border-radius: 0ex ;
2272     border: none ;
2273 }
2274
2275
2276
2277 nav.sidetoc {
2278     font-family: "DejaVu Serif", "Bitstream Vera Serif",
2279         "Lucida Bright", Georgia, serif;
2280     float:right ;
2281     width: 20%;
2282     border-left: 1px solid silver;
2283     border-top: 1px solid silver;
2284     border-bottom: 1px solid silver;
2285 /*     border-top: 2px solid #808080 ; */
2286     background: #FAF7F4 ;
2287     padding: 2ex 0em 2ex 1em ;
2288     margin: 0ex 0em 2ex 1em ;
2289     font-size:.9em ;
2290     border-radius: 20px 0px 0px 20px ;
2291 }
2292
2293 div.sidetoccontents {
2294 /*     border-top: 1px solid silver ; */
2295     overflow-y: auto ;
2296     width: 100% ;
2297     text-align: left ;
```

```
2298 }
2299
2300
2301 nav.sidetoc p {line-height:1.2 ; margin: 1ex .5em 1ex .5em ;
2302     text-indent: 0 ; }
2303
2304 nav.sidetoc p a {color:black ; font-size: .7em ;}
2305
2306 div.sidetoc title {font-size: 1.2em; font-weight:bold; text-align:center;
2307     border-bottom: 1px solid silver ;    }
2308
2309 nav.sidetoc a:hover {text-decoration: underline ; }
2310
2311
2312
2313 section.textbody { margin: 0ex 1em 0ex 1em ;}
2314
2315
2316 div.multicolsheading { -webkit-column-span: all;
2317     -moz-column-span: all; column-span: all; }
2318 div.multicols { -webkit-columns: 3 380px ;
2319     -moz-columns: 3 380px ; columns: 3 380px ; }
2320 div.multicols p {margin-top: 0ex}
2321
2322
2323 /* Used for algorithm2e: */
2324 div.alg2evline{
2325     margin-left: 1em ;
2326     padding-left: 1em ;
2327     border-left: 1px solid black ;
2328     border-radius: 0px 0px 0px 1ex ;
2329 }
2330
2331 div.alg2evsline{
2332     margin-left: 1em ;
2333     padding-left: 1em ;
2334     border-left: 1px solid black ;
2335 }
2336
2337 div.alg2enoline{
2338     margin-left: 1em ;
2339     padding-left: 1em ;
2340 }
2341
2342 span.alg2elinenumber{
2343     margin-right: .5em ;
2344     font-size: 50% ;
2345     color: red ;
2346 }
2347
```

```
2348
2349 /* Used for algorithmicx: */
2350 span.floatright { float: right ; }
2351
2352
2353
2354
2355 /* Native LaTeX theorems: */
2356
2357 .theoremcontents { font-style: italic; margin-top: 3ex ; margin-bottom: 3ex ; }
2358 .theoremlabel { font-style: normal; font-weight: bold ; margin-right: .5em ; }
2359
2360
2361 /* theorem, amsthm, and ntheorem packages */
2362
2363 span.theoremheader,
2364 span.theoremheaderplain,
2365 span.theoremheaderdefinition,
2366 span.theoremheaderbreak,
2367 span.theoremheadermarginbreak,
2368 span.theoremheaderchangebreak,
2369 span.theoremheaderchange,
2370 span.theoremheadermargin
2371 {
2372     font-style:normal ; font-weight: bold ; margin-right: 1em ;
2373 }
2374
2375 span.amsthmnameplain,
2376 span.amsthmnamedefinition,
2377 span.amsthmnumberplain,
2378 span.amsthmnumberdefinition
2379 {
2380     font-style:normal ; font-weight: bold ;
2381 }
2382
2383
2384 span.amsthmnameremark,
2385 span.amsthmnumberremark
2386 {font-style:italic ; font-weight: normal ; }
2387
2388
2389 span.amsthmnoteplain,
2390 span.amsthmnotedefinition
2391 {font-style:normal ;}
2392
2393
2394 span.theoremheaderremark,
2395 span.theoremheaderproof,
2396 span.amsthmproofname
2397 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
```

```
2398
2399 span.theoremheadersc
2400 {
2401     font-style:normal ;
2402     font-variant: small-caps ;
2403     font-weight: normal ;
2404     margin-right: 1em ;
2405 }
2406
2407 .theoremdemark {float:right}
2408
2409 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
2410 div.theorembodybreak, div.theorembodynonumberbreak,
2411 div.theorembodymarginbreak,
2412 div.theorembodychangebreak,
2413 div.theorembodychange,
2414 div.theorembodymargin
2415 {
2416     font-style:italic;
2417     margin-top: 3ex ; margin-bottom: 3ex ;
2418 }
2419
2420 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
2421 div.theorembodyplainupright, nonumberplainuprightsc,
2422 div.amsthmbodydefinition, div.amsthmbodyremark,
2423 div.amsthmproof
2424 {
2425     font-style: normal ;
2426     margin-top: 3ex ; margin-bottom: 3ex ;
2427 }
2428
2429 span.amsthmnoteremark {}
2430
2431
2432
2433 /*
2434 For CSS LaTeX and related logos:
2435 Based on:
2436 http://edward.oconnor.cx/2007/08/tex-poshlet
2437 http://nitens.org/taraborelli/texlogo
2438 */
2439
2440 .latexlogofont {
2441     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2442         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2443     font-variant: normal ;
2444 }
2445
2446 .latexlogo {
2447     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
```

```
2448     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2449     letter-spacing: .03em ;
2450     font-size: 1.1em;
2451 }
2452
2453 .latexlogo sup {
2454     text-transform: uppercase;
2455     letter-spacing: .03em ;
2456     font-size: 0.85em;
2457     vertical-align: 0.15em;
2458     margin-left: -0.36em;
2459     margin-right: -0.15em;
2460 }
2461
2462 .latexlogo sub {
2463     text-transform: uppercase;
2464     vertical-align: -0.5ex;
2465     margin-left: -0.1667em;
2466     margin-right: -0.125em;
2467     font-size: 1em;
2468 }
2469
2470 .xetexlogo {
2471     font-family: "Linux Libertine O", "Nimbus Roman No 9 L",
2472     "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2473     letter-spacing: .03em ;
2474     font-size: 1.1em;
2475 }
2476
2477 /* A smaller gap between Xe and Tex v.s. LaTeX: */
2478 .xetexlogo sub {
2479     text-transform: uppercase;
2480     vertical-align: -0.5ex;
2481     margin-left: -0.0667em;
2482     margin-right: -0.2em;
2483     font-size: 1em;
2484     letter-spacing: .03em ;
2485 }
2486
2487 /* A large gap between Xe and LaTeX v.s. TeX: */
2488 .xelatexlogo sub {
2489     text-transform: uppercase;
2490     vertical-align: -0.5ex;
2491     margin-left: -0.0667em;
2492     margin-right: -.05em;
2493     font-size: 1em;
2494     letter-spacing: .03em ;
2495 }
2496
2497 .amslogo {
```

```
2498     font-family: "TeXGyreChorus","URW Chancery L",
2499         "Apple Chancery","ITC Zapf Chancery","Monotype Corsiva",
2500         "Linux Libertine O", "Nimbus Roman No 9 L", "FreeSerif",
2501         "Hoefler Text", Times, "Times New Roman", serif;
2502     font-style: italic;
2503 }
2504
2505 .lyxlogo {
2506     font-family: "URW Classico", Optima, "Linux Biolinum O",
2507         "DejaVu Sans", "Bitstream Vera Sans", Geneva,
2508         Verdana, sans-serif ;
2509 }
2510
2511
2512 /* Only display top and bottom navigation if a small screen: */
2513 /* Hide the sidetoc if a small screen: */
2514 nav.topnavigation { display:none; }
2515 nav.botnavigation { display:none; }
2516
2517 @media screen and (max-width: 45em) {
2518     /*     nav.sidetoc {display:none;} */
2519     nav.sidetoc {
2520         float: none ;
2521         width: 100% ;
2522         margin: 5ex 0px 5ex 0px ;
2523         padding: 0 ;
2524         border-radius: 0 ;
2525         border-bottom: 1px solid black ;
2526         border-top: 1px solid black ;
2527         box-shadow: none ;
2528     }
2529     /*     nav.topnavigation { display:block } */
2530     nav.botnavigation { display:block }
2531     .marginpar {
2532         max-width: 100%;
2533         float: none;
2534         display:block ;
2535         margin: 1ex 1em 1ex 1em ;
2536     }
2537 }
2538
2539 @media print {
2540     body {
2541         font-family: "Linux Libertine O",
2542             "DejaVu Serif", "Bitstream Vera Serif",
2543             "Liberation Serif", "Nimbus Roman No 9 L",
2544             "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2545     }
2546     nav.sidetoc { display:none; }
2547     nav.topnavigation { display: none; }
```

```

2548     nav.botnavigation { display: none; }
2549 }
2550
2551 @media handheld {
2552     nav.sidetoc { display:none; }
2553     nav.topnavigation { display:block }
2554     nav.botnavigation { display:block }
2555 }
2556
2557 @media projection {
2558     nav.sidetoc { display:none; }
2559     nav.topnavigation { display:block }
2560     nav.botnavigation { display:block }
2561 }
2562 \end{filecontents*}
2563 % \end{Verbatim}% for syntax highlighting
2564 \end{warpprint}

```

37.5 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.

```

2565 \begin{warpprint}
2566 \begin{filecontents*}{lwarp_sagebrush.css}
2567 @import url("lwarp.css") ;
2568
2569
2570 A:link {color:#105030 ; text-decoration: none ; }
2571 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2572 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2573 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}
2574
2575
2576
2577 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2578 {
2579     font-family: "URW Classico", Optima, "Linux Biolinum O",
2580         "Linux Libertine O", "Liberation Serif",
2581         "Nimbus Roman No 9 L", "FreeSerif",
2582         "Hoefler Text", Times, "Times New Roman", serif;
2583     font-variant: small-caps ;
2584     font-weight: normal ;
2585     color: #304070 ;
2586     text-shadow: 2px 2px 3px #808080;

```

```
2587 }
2588
2589 h1 { /* title of the entire website, used on each page */
2590     font-variant: small-caps ;
2591     color: #304070 ;
2592     text-shadow: 2px 2px 3px #808080;
2593     background-color: #F7F7F0 ;
2594     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2595 }
2596
2597 h1 {
2598     border-bottom: 1px solid #304070;
2599     border-top: 2px solid #304070;
2600 }
2601
2602 h2 {
2603     border-bottom: 1px solid #304070;
2604     border-top: 2px solid #304070;
2605     background-color: #F7F7F0 ;
2606     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2607 }
2608
2609
2610
2611 div.abstract {
2612     background: #f5f5eb ;
2613     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2614
2615     border: 1px solid silver;
2616     border-radius: 1em ;
2617 }
2618
2619 div.abstract dl {line-height:1.5;}
2620 div.abstract dt {color:#304070;}
2621
2622 div.abstracttitle{
2623     font-family: "URW Classico", Optima, "Linux Biolinum O",
2624         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2625         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2626     font-weight:bold;
2627     font-variant: small-caps ;
2628     font-size:1.5em;
2629     border-bottom: 1px solid silver ;
2630     color: #304070 ;
2631     text-align: center ;
2632     text-shadow: 1px 1px 2px #808080;
2633 }
2634
2635 span.abstracrunintitle{
2636     font-family: "URW Classico", Optima, "Linux Biolinum O",
```

```
2637         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2638         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2639     font-weight:bold;
2640 }
2641
2642
2643 div.epigraph, div.dictum {
2644     background: #f5f5eb ;
2645     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2646
2647     border: 1px solid silver ;
2648     border-radius: 1ex ;
2649     box-shadow: 3px 3px 3px #808080 ;
2650 }
2651
2652
2653 .example {
2654     background-color: #f5f5eb ;
2655     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2656
2657 }
2658
2659 div.exampletitle{
2660     font-family: "URW Classico", Optima, "Linux Biolinum O",
2661         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2662         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2663     font-weight:bold;
2664     font-variant: small-caps ;
2665     border-bottom: 1px solid silver ;
2666     color: #304070 ;
2667     text-align: center ;
2668     text-shadow: 1px 1px 2px #808080;
2669 }
2670
2671
2672 .sidebar {
2673     background-color: #f5f5eb ;
2674     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2675
2676 }
2677
2678 div.sidebar{
2679     font-family: "URW Classico", Optima, "Linux Biolinum O",
2680         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2681         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2682     font-weight:bold;
2683     font-variant: small-caps ;
2684     border-bottom: 1px solid silver ;
2685     color: #304070 ;
2686     text-align: center ;
```

```
2687     text-shadow: 1px 1px 2px #808080;
2688 }
2689
2690
2691 .fancyvrblabel {
2692     font-family: "URW Classico", Optima, "Linux Biolinum O",
2693         "Linux Libertine O", "Liberation Serif", "Nimbus Roman No 9 L",
2694         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2695     font-weight:bold;
2696     font-variant: small-caps ;
2697     font-size: 1.5em ;
2698     color: #304070 ;
2699     text-align: center ;
2700     text-shadow: 1px 1px 2px #808080;
2701 }
2702
2703 div.minipage {
2704     background-color: #eeeeee7 ;
2705     border: 1px solid silver ;
2706     border-radius: 1ex ;
2707 }
2708
2709 table div.minipage { background: none ; border: none ; }
2710
2711 div.framebox div.minipage {border:none ; background:none}
2712
2713 section.textbody > div.minipage {
2714     box-shadow: 3px 3px 3px #808080 ;
2715 }
2716
2717 div.fboxBlock div.minipage { box-shadow: none ; }
2718
2719 .framed .minipage , .framedleftbar .minipage {
2720     border: none ;
2721     background: none ;
2722     padding: 0ex ;
2723     margin: 0ex ;
2724 }
2725
2726 figure.figure .minipage, figcaption .minipage { border: none; }
2727
2728 div.marginblock div.minipage ,
2729 div.marginparblock div.minipage
2730     { border: none; }
2731
2732 figure , div.marginblock {
2733     background-color: #eeeeee7 ;
2734     border: 1px solid silver ;
2735     border-radius: 1ex ;
2736     box-shadow: 3px 3px 3px #808080 ;
```

```
2737 }
2738
2739 figure figure {
2740     border: 1px solid silver ;
2741     margin: 0em ;
2742     box-shadow: none ;
2743 }
2744
2745 /*
2746 figcaption {
2747     border-top: 1px solid silver ;
2748     border-bottom: 1px solid silver ;
2749     background-color: #e8e8e8 ;
2750 }
2751 */
2752
2753
2754 div.table {
2755     box-shadow: 3px 3px 3px #808080 ;
2756 }
2757
2758 /*
2759 .tnotes {
2760     background: #e8e8e8;
2761     border: 1px solid silver;
2762 }
2763 */
2764
2765
2766 nav.topnavigation{
2767     background-color: #b0b8b0 ;
2768     background-image: linear-gradient(to bottom,#e0e0e0,#b0b8b0) ;
2769 }
2770
2771 nav.botnavigation{
2772     background-color: #b0b8b0 ;
2773     background-image: linear-gradient(to top,#e0e0e0,#b0b8b0) ;
2774 }
2775
2776
2777
2778 header{
2779     background-color: #F7F7F0 ;
2780     background-image: linear-gradient(to top, #F7F7F0, #b0b8b0);
2781 }
2782
2783 footer{
2784     background-color: #F7F7F0 ;
2785     background-image: linear-gradient(to bottom, #F7F7F0, #b0b8b0);
2786 }
```

```

2787
2788
2789
2790 nav.sidetoc {
2791     background-color: #F7F7F0 ;
2792     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C0);
2793     box-shadow: 3px 3px 3px #808080 ;
2794     border-radius: 0px 0px 0px 20px ;
2795 }
2796
2797 div.sidetoc title {color: #304070 ; }
2798
2799 nav.sidetoc a: hover {
2800     color: #006000 ;
2801     text-decoration: none ;
2802     text-shadow: 0px 0px 2px #a0a0a0;
2803 }
2804
2805
2806 @media screen and (max-width: 45em) {
2807     nav.sidetoc { border-radius: 0 ; }
2808 }
2809
2810
2811 \end{filecontents*}
2812 % \end{Verbatim}% for syntax highlighting
2813 \end{warpprint}

```

37.6 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.

```

2814 \begin{warpprint}
2815 \begin{filecontents*}{lwarp_formal.css}
2816 @import url("lwarp.css") ;
2817
2818
2819
2820 A:link {color: #802020 ; text-decoration: none; }
2821 A:visited {color: #802020 ; text-shadow: none ;}
2822 A: hover {color: #400000 ; text-shadow: none ;}
2823 A: active {color: #C00000 ; text-shadow: none ;}
2824
2825

```

```
2826 body {
2827     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2828         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2829         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2830         "Times New Roman", serif;
2831     background: #fffcf5;
2832 }
2833
2834 span.textrm {
2835     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2836         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2837         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2838         "Times New Roman", serif;
2839 }
2840
2841 span.textsf {
2842     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2843         Geneva, Verdana, sans-serif ;
2844 }
2845
2846
2847
2848 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2849 {
2850     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2851         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2852         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2853         "Times New Roman", serif;
2854     color: #800000 ;
2855     text-shadow: none ;
2856 }
2857
2858 h1, h2 {
2859     background-color: #fffcf5 ;
2860     background-image: none ;
2861     border-bottom: 1px solid #808080;
2862     border-top: 2px solid #808080;
2863 }
2864
2865 div.abstracttitle {
2866     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2867         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2868         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2869         "Times New Roman", serif;
2870     color: black ;
2871     text-shadow: none ;
2872 }
2873
2874 span.abstrctruintitle {
2875     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
```

```
2876         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2877         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2878         "Times New Roman", serif;
2879     color: black ;
2880     text-shadow: none ;
2881 }
2882
2883 div.abstract { font-size: 100% }
2884
2885 .sidebar {
2886     background: #fffcf5;
2887     background-image: none ;
2888     margin: 2em 5% 2em 5%;
2889     padding: 0.5em 1em;
2890     border: none ;
2891     border-top : 1px solid silver;
2892     border-bottom : 1px solid silver;
2893     font-size: 90% ;
2894 }
2895
2896 div.sidebartitle{
2897     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2898         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2899         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2900         "Times New Roman", serif;
2901     color: #800000 ;
2902     text-shadow: none ;
2903     border: none ;
2904 }
2905
2906 .example {
2907     background: #fffcf5;
2908     background-image: none ;
2909     margin: 2em 5% 2em 5%;
2910     padding: 0.5em 1em;
2911     border: none ;
2912     border-top : 1px solid silver;
2913     border-bottom : 1px solid silver;
2914 }
2915
2916 div.exampletitle{
2917     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2918         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2919         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2920         "Times New Roman", serif;
2921     color: #800000 ;
2922     text-shadow: none ;
2923     border: none ;
2924 }
2925
```

```
2926 div.fancyvrblabel{
2927     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2928         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2929         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2930         "Times New Roman", serif;
2931     color: #800000 ;
2932     text-shadow: none ;
2933     border: none ;
2934 }
2935
2936
2937
2938 .verse {
2939     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2940         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2941         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2942         "Times New Roman", serif;
2943 }
2944
2945
2946 figure {
2947     margin: 5ex 5% 5ex 5% ;
2948     padding: 1ex 1em 1ex 1em ;
2949     background-color: #fffcf5 ;
2950     overflow-x: auto ;
2951     border: none ;
2952 /*     border-top: 1px solid silver; */
2953 /*     border-bottom: 1px solid silver; */
2954 }
2955
2956
2957 figcaption , .lstlisting {
2958     border: none ;
2959 /*     border-top: 1px solid silver ; */
2960 /*     border-bottom: 1px solid silver ; */
2961     background-color: #fffcf5 ;
2962 }
2963
2964 .tnotes {
2965     background: #fffcf5 ;
2966 }
2967
2968 .theorem {
2969     background: none ;
2970 }
2971
2972 .minipage {
2973     background-color: #fffcf5 ;
2974     border: none ;
2975 }
```

```
2976
2977 div.floatrow figure { border: none ; }
2978
2979 figure figure { border: none ; }
2980
2981
2982 nav.toc, nav.lof, nav.lot, nav.lol {
2983     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2984         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2985         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2986         "Times New Roman", serif;
2987 }
2988
2989 nav.sidetoc {
2990     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2991         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2992         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2993         "Times New Roman", serif;
2994     background-image: linear-gradient(to bottom, #fffcf5, #C0C0C0);
2995     border-radius: 0px 0px 0px 20px ;
2996 }
2997
2998 div.sidetoctitle{
2999     color: #800000 ;
3000 }
3001
3002 header{
3003     background-color: #e0e0e0 ;
3004     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
3005     text-align:center ;
3006 }
3007
3008 footer{
3009     background-color: #e0e0e0 ;
3010     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
3011     padding: 2ex 1em 2ex 1em ;
3012     clear:right ;
3013     text-align:left ;
3014 }
3015
3016 nav.botnavigation {
3017     background: #dedcd5 ;
3018     border-top: 1px solid black ;
3019 }
3020 \end{filecontents*}
3021 % \end{Verbatim}% for syntax highlighting
3022 \end{warpprint}
```

37.7 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.

```

3023 \begin{warpprint}
3024 \begin{filecontents*}{sample_project.css}
3025 /* ( --- Start of project.css --- ) */
3026 /* ( --- A sample project-specific CSS file for lwarp --- ) */
3027
3028 /* Uncomment one of the following: */
3029 @import url("lwarp.css") ;
3030 /* @import url("lwarp_formal.css") ; */
3031 /* @import url("lwarp_sagebrush.css") ; */
3032
3033 /* Project-specific CSS setting follow here. */
3034 /* . . . */
3035
3036 /* ( --- End of project.css --- ) */
3037 \end{filecontents*}
3038 % \end{Verbatim}% for syntax highlighting
3039 \end{warpprint}

```

37.8 lwarp.xdy

File `lwarp.xdy` Used to modify the index for `lwarp`.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```

3040 \begin{warpprint}
3041 \begin{filecontents*}{lwarp.xdy}
3042 (require "tex/inputenc/latin.xdy")
3043 (merge-rule "\\PS *" "Postscript")
3044 (require "texindy.xdy")
3045 (require "page-ranges.xdy")
3046 (require "book-order.xdy")
3047 (require "page-ranges.xdy")
3048 (markup-locref :open "\hyperindexref{" :close "}")
3049 (define-location-class "arabic-page-numbers"
3050   ("arabic-numbers") :min-range-length 1)
3051 (define-location-class-order ("roman-page-numbers"
3052   "arabic-page-numbers"
3053   "alpha-page-numbers"

```

```

3054             "Roman-page-numbers"
3055             "Alpha-page-numbers"
3056             "see"
3057             "seealso"))
3058 \end{filecontents*}
3059 % \end{Verbatim}% for syntax highlighting
3060 \end{warpprint}

```

37.9 lwarp_one_limage.cmd

File `lwarp_one_limage.cmd` Used by **lwarp** to help make lateximages when using WINDOWS.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

The arguments are each of the three fields from `lateximages.txt`, and also the base name of the source file.

MikTeX does not allow file `lwarp_one_limage.cmd` to be created directly by **lwarpmk**, so `lwarp_one_limage.txt` is created instead, then copied to `lwarp_one_limage.cmd` by **lwarpmk**. This occurs each time **lwarpmk** used to create lateximages.

```

3061 \begin{warpprint}
3062 \begin{filecontents*}{lwarp_one_limage.txt}
3063 @echo off
3064 pdfseparate -f %1 -l %1 %4_html.pdf lateximages\lateximagetemp-%d.pdf
3065 pdfcrop -- hires lateximages\lateximagetemp-%1.pdf lateximages\%3.pdf
3066 pdftocairo -svg -noshrink lateximages\%3.pdf lateximages\%3.svg
3067 del lateximages\%3.pdf
3068 del lateximages\lateximagetemp-%1.pdf
3069 exit
3070 \end{filecontents*}
3071 \end{warpprint}

```

37.10 lwarp_mathjax.txt

File `lwarp_mathjax.txt` Used by **lwarp** when using MATHJAX.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```

3072 \begin{warpprint}
3073 \begin{filecontents*}{lwarp_mathjax.txt}
3074 <!-- https://groups.google.com/forum/#!topic/

```

```

3075                                     mathjax-users/jUtewUcE2bY -->
3076 <script type="text/x-mathjax-config">
3077 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
3078     var seteqsectionDefault = {name: "", num: 0};
3079     var seteqsections = {}, seteqsection = seteqsectionDefault;
3080     var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
3081     var AMS = MathJax.Extension["TeX/AMSmath"];
3082     TEX.Definitions.Add({
3083     macros: {
3084         seteqsection: "mySection",
3085         seteqnumber: "mySetEqNumber"
3086     }
3087     });
3088
3089     PARSE.Augment({
3090     mySection: function (name) {
3091         seteqsection.num = AMS.number;
3092         var n = this.GetArgument(name);
3093         if (n === "") {
3094             seteqsection = seteqsectionDefault;
3095         } else {
3096             if (!seteqsections["_"+n])
3097                 seteqsections["_"+n] = {name:n, num:0};
3098             seteqsection = seteqsections["_"+n];
3099         }
3100         AMS.number = seteqsection.num;
3101     },
3102     mySetEqNumber: function (name) {
3103         var n = this.GetArgument(name);
3104         if (!n || !n.match(/^ *[0-9]+ *$/))
3105             n = ""; else n = parseInt(n)-1;
3106         <!-- $ syntax highlighting -->
3107         if (n === "" || n < 1)
3108             TEX.Error
3109             ("Argument to "+name+" should be a positive integer");
3110         AMS.number = n;
3111     }
3112     });
3113     MathJax.Hub.Config({
3114     TeX: {
3115         equationNumbers: {
3116         formatTag: function (n)
3117             {return "("+(seteqsection.name+"."+n).replace(/\./,"")+")"},
3118         formatID: function (n) {
3119             n = (seteqsection.name+'.'+n).replace
3120             (/[<>']/g,"").replace(/\./,"");
3121             return 'mjax-eqn-' + n;
3122         }
3123         }
3124     }

```

```

3125     });
3126 });
3127 </script>
3128
3129 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
3130 <script type="text/x-mathjax-config">
3131   MathJax.Ajax.config.path["Contrib"] =
3132     "https://cdn.mathjax.org/mathjax/contrib";
3133 </script>
3134
3135 <!-- https://github.com/burnpanck/MathJax-siunitx -->
3136
3137 <script type="text/x-mathjax-config">
3138   MathJax.Hub.Config({
3139     extensions: ["tex2jax.js", "[siunitx]/siunitx.js"],
3140     jax: ["input/TeX", "output/HTML-CSS"],
3141     tex2jax: {
3142       inlineMath: [["$","$"],["\\(","\\)"]],
3143       processClass: "tabbing|verse"
3144     },
3145     TeX: {extensions: ["AMSMath.js", "AMSsymbols.js", "sinuitx.js"]}
3146   });
3147   MathJax.Ajax.config.path['siunitx'] = 'http://rawgit.com/burnpanck/MathJax-siunitx/master/';
3148 </script>
3149
3150 <script type="text/x-mathjax-config">
3151   MathJax.Hub.Config({
3152     TeX: {
3153       equationNumbers: {
3154         autoNumber: "AMS"
3155       }
3156     }
3157   });
3158 </script>
3159
3160 <!-- Alternative CDN provider: -->
3161 <script type="text/javascript" async
3162   src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_HTML-full">
3163 </script>
3164
3165 <!-- No longer supported after April 30, 2017: -->
3166 <!--
3167 <script
3168   src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS_HTML-full">
3169 </script>
3170 -->
3171
3172 \end{filecontents*}
3173 % \end{Verbatim}% for syntax highlighting
3174 \end{warpprint}

```

37.11 lwarpmk.lua — lwarpmk option

Opt `lwarpmk` Creates a local copy of **lwarpmk**.

Prog `lwarpmk` Command-line utility to process **lwarp** files and images.

[parallel processing](#) `lateximages` and `svg` math images are generated using multiple processes in parallel. For UNIX and LINUX, every 32 images the `wait` command is issued to wait for the previous batch of images to finish processing before starting a new batch. For WINDOWS, every 32 images one task is dispatched with

```
START /B /WAIT /BELOWNORMAL
```

which causes the operating system to wait until this lesser-priority tasks finishes, hopefully also waiting for the normal priority tasks which were already in progress to also complete. Afterwards, the next batch of images is started.

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

```
3175 \begin{LWR@createlwarpmk}

3176 \begin{filecontents*}{lwarpmk.lua}
3177 #!/usr/bin/env texlua
3178
3179 -- Copyright 2016-2018 Brian Dunn
3180
3181
3182 printversion = "v0.57"
3183
3184
3185 function printhelp ()
3186 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
3187 end
3188
3189
3190 function printusage ()
3191 --
3192 -- Print the usage of the lwarpmk command:
3193 --
3194 print ( [[
3195
3196 lwarpmk print [project]: Compile the print version if necessary.
3197 lwarpmk print1 [project]: Forced single compile of the print version.
3198 lwarpmk printindex [project]: Process the index for the print version.
3199 lwarpmk printglossary [project]: Process the glossary for the print version.
3200 lwarpmk html [project]: Compile the HTML version if necessary.
3201 lwarpmk html1 [project]: Forced single compile of the HTML version.
3202 lwarpmk htmlindex [project]: Process the index for the html version.
3203 lwarpmk htmlglossary [project]: Process the glossary for the html version.
```

```

3204 lwarpmk again [project]: Touch the source code to trigger recompiles.
3205 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
3206 lwarpmk pdftohtml [project]:
3207     For use with latexmk or a Makefile:
3208     Converts project_html.pdf to project_html.html and individual HTML files.
3209     Finishes the HTML conversion even if there was a compile error.
3210 lwarpmk pdftosvg <list of file names>: Converts each PDF file to SVG.
3211 lwarpmk clean [project]: Remove .aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
3212 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
3213 lwarpmk cleanlimages: Removes all images from the "lateximages" directory.
3214 lwarpmk -h: Print this help message.
3215 lwarpmk --help: Print this help message.
3216
3217 ]] )
3218 printconf ()
3219 end
3220
3221
3222 function printconf ()
3223 --
3224 -- Print the format of the configuration file lwarpmk.conf:
3225 --
3226 print ( [[
3227 An example lwarpmk.conf or <project>.lwarpmkconf project file:
3228 --
3229 opsystem = "Unix"      (or "Windows")
3230 latexname = "pdflatex" (or "lualatex", or "xelatex")
3231 sourcename = "projectname" (the source-code filename w/o .tex)
3232 homehtmlfilename = "index" (or perhaps the project name)
3233 htmlfilename = "" (or "projectname" - filename prefix)
3234 latexmk = "false" (or "true" to use latexmk to build PDFs)
3235 shellescape = "false"
3236 xindylanguage = "english" (use a language supported by xindy)
3237 xindycodepage = "utf8" (use a codepage supported by xindy)
3238 xindystyle = "lwarp.xdy" (or a custom file based on lwarp.xdy)
3239 pdftotextenc = "UTF-8" (use an encoding supported by pdftotext)
3240 --
3241 Filenames must contain only letters, numbers, underscore, or dash.
3242 Values must be in upright "quotes".
3243
3244 ]] ) ;
3245 end
3246
3247
3248
3249 function splitfile (destfile,sourcefile)
3250 --
3251 -- Split one large sourcefile into a number of files,
3252 -- starting with destfile.
3253 -- The file is split at each occurrence of <!--|Start file|newfilename|*

```

```
3254 --
3255 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
3256 local sfile = io.open(sourcefile)
3257 io.output(destfile)
3258 for line in sfile:lines() do
3259   i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
3260   if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then
3261     -- split the file
3262     io.output(newfilename) ;
3263   else
3264     -- not a splitpoint
3265     io.write (line .. "\n") ;
3266   end
3267 end -- do
3268 io.close(sfile)
3269 end -- function
3270
3271
3272 function cvalueerror ( line, linenum , cvalue )
3273 --
3274 -- Incorrect value, so print an error and exit.
3275 --
3276   print ("lwarpmk: ===")
3277   print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3278   print (
3279     "lwarpmk: incorrect variable value \"" .. cvalue ..
3280     "\" in lwarpmk.conf.\n"
3281   ) ;
3282   print ("lwarpmk: ===")
3283   printconf () ;
3284   os.exit(1) ;
3285 end
3286
3287
3288 function loadconf ()
3289 --
3290 -- Load settings from the project's "lwarpmk.conf" file:
3291 --
3292 -- Default configuration filename:
3293 local conffile = "lwarpmk.conf"
3294 -- Optional configuration filename:
3295 if ( arg[2] ~= nil ) then conffile = arg[2].."lwarpmkconf" end
3296 -- Additional defaults:
3297 opsystem = "Unix"
3298 latexmk = "false"
3299 shellescape = "false"
3300 xindylanguage = "english"
3301 xindycodepage = "utf8"
3302 xindystyle = "lwarp.xdy"
3303 pdftotextenc = "UTF-8"
```

```

3304-- Verify the file exists:
3305 if (lfs.attributes(conffile,"mode")==nil) then
3306     -- file not exists
3307     print ("lwarpmk: ===")
3308     print ("lwarpmk: File \" .. conffile ..\" does not exist.")
3309     print ("lwarpmk: Move to the project's source directory,")
3310     print ("lwarpmk: recompile using pdflatex, xelatex, or lualatex,")
3311     print ("lwarpmk: then try using lwarpmk again.")
3312     if ( arg[2] ~= nil ) then
3313         print (
3314             "lwarpmk: (\" .. arg[2] ..
3315             "\" does not appear to be a project name.)"
3316         )
3317     end
3318     print ("lwarpmk: ===")
3319     printhelp () ;
3320     os.exit(1) -- exit the entire lwarpmk script
3321 else -- file exists
3322 -- Read the file:
3323 print ("lwarpmk: Reading " .. conffile .. ".")
3324 local cfile = io.open(conffile)
3325 -- Scan each line:
3326 local linenum = 0
3327 for line in cfile:lines() do -- scan lines
3328     linenum = linenum + 1
3329     i,j,cvarname,cvalue = string.find (line,"([%w-_]*)%s*=%s*\"([%w%-_.]*)\"") ;
3330 -- Error if incorrect enclosing characters:
3331 if ( i == nil ) then
3332     print ("lwarpmk: ===")
3333     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3334     print ("lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
3335     print ("lwarpmk: ===")
3336     printconf () ;
3337     os.exit(1) ;
3338 end -- nil
3339 if ( cvarname == "opsystem" ) then
3340     -- Verify choice of opsystem:
3341     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
3342         opsystem = cvalue
3343     else
3344         cvalueerror ( line, linenum , cvalue )
3345     end
3346 elseif ( cvarname == "latexname" ) then
3347     -- Verify choice of LaTeX compiler:
3348     if (
3349         (cvalue == "pdflatex") or
3350         (cvalue == "xelatex") or
3351         (cvalue == "lualatex")
3352     ) then
3353         latexname = cvalue

```

```

3354     else
3355         cvalueerror ( line, linenum , cvalue )
3356     end
3357 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
3358 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
3359 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
3360 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
3361 elseif ( cvarname == "shellescape" ) then shellescape = cvalue
3362 elseif ( cvarname == "xindylanguage" ) then xindylanguage = cvalue
3363 elseif ( cvarname == "xindycodepage" ) then xindycodepage = cvalue
3364 elseif ( cvarname == "xindystyle" ) then xindystyle = cvalue
3365 elseif ( cvarname == "pdftotextenc" ) then pdftotextenc = cvalue
3366 else
3367     print ("lwarpmk: ===")
3368     print ("lwarpmk: " .. linenum .. " : " .. line ) ;
3369     print (
3370         "lwarpmk: Incorrect variable name \" " .. cvarname .. "\" in " ..
3371         conffile .. ".\n"
3372     ) ;
3373     print ("lwarpmk: ===")
3374     printconf () ;
3375 os.exit(1) ;
3376 end -- cvarname
3377 end -- do scan lines
3378 io.close(cfile)
3379 end -- file exists
3380 -- Error if sourcename is "lwarp".
3381 -- This could happen if a local copy of lwarp has recently been recompiled.
3382 if sourcename=="lwarp" then
3383     print ("lwarpmk: ===")
3384     print ("lwarpmk: Lwarp has recently been recompiled in this directory,")
3385     print ("lwarpmk: and \"lwarpmk.conf\" is no longer set for your own project.")
3386     print ("lwarpmk: Recompile your own project using pdf/luaxelatex <projectname>.")
3387     print ("lwarpmk: After a recompile, \"lwarpmk.conf\" will be set for your project,")
3388     print ("lwarpmk: and you may again use lwarpmk.")
3389     print ("lwarpmk: ===")
3390     os.exit(1)
3391 end -- sourcename of "lwarp"
3392 -- Select some operating-system commands:
3393 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
3394     rmname = "rm"
3395     mvname = "mv"
3396     cpname = "cp"
3397     touchnamepre = "touch"
3398     touchnamepost = ""
3399     newtouchname = "touch"
3400     dirslash = "/"
3401     opquote= "\"'"
3402     cmdgroupopenname = " ( "
3403     cmdgroupclosename = " ) "

```

```

3404     seqname = " && "
3405     bgnname = " &"
3406 elseif opsystem=="Windows" then -- For Windows
3407     rmname = "DEL"
3408     mvname = "MOVE"
3409     cpname = "COPY"
3410     touchnamepre = "COPY /b"
3411     touchnamepost = "+,,"
3412     newtouchname = "echo empty >"
3413     dirslash = "\\\"
3414     opquote= "\""
3415     cmdgroupopenname = ""
3416     cmdgroupclosename = ""
3417     seqname = " & "
3418     bgnname = ""
3419 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
3420 end --- for Windows
3421
3422 -- set xindycmd, glossarycmd according to pdflatex vs xelatex/lualatex:
3423 if ( latexname == "pdflatex" ) then
3424     xindycmd = "texindy "
3425     glossarycmd = "xindy "
3426 else
3427     xindycmd = "xindy -M texindy "
3428     glossarycmd = "xindy "
3429 end
3430
3431 end -- loadconf
3432
3433
3434 function executecheckerror ( executecommands , errormessage )
3435 --
3436 -- Execute an operating system call,
3437 -- and maybe exit with an error message.
3438 --
3439 local err
3440 err = os.execute ( executecommands )
3441 if ( err ~= 0 ) then
3442     print ("lwarpmk: ===")
3443     print ("lwarpmk: " .. errormessage )
3444     print ("lwarpmk: ===")
3445     os.exit(1)
3446 end
3447 end -- executecheckerror
3448
3449
3450 function refreshdate ()
3451 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
3452 end
3453

```

```
3454
3455
3456 function reruntoget (filesourc)
3457 --
3458 -- Scan the LaTeX log file for the phrase "Rerun to get",
3459 -- indicating that the file should be compiled again.
3460 -- Return true if found.
3461 --
3462 local fsourc = io.open(filesourc)
3463 for line in fsourc:lines() do
3464 if ( string.find(line,"Rerun to get") ~= nil ) then
3465     io.close(fsourc)
3466     return true
3467 end -- if
3468 end -- do
3469 io.close(fsourc)
3470 return false
3471 end
3472
3473
3474
3475 function onetime (fsuffix)
3476 --
3477 -- Compile one time, return true if should compile again.
3478 -- fsuffix is "" for print, "_html" for HTML output.
3479 --
3480 print("lwarpmk: Compiling with " .. latexname .. " " " " .. sourcename..fsuffix)
3481 local thisshellescape = " "
3482 if ( shellescape == "true" ) then
3483     thisshellescape = " -shell-escape "
3484 else
3485     thisshellescape = " "
3486 end
3487 executecheckerror (
3488     latexname .. thisshellescape .. sourcename..fsuffix ,
3489     "Compile error."
3490 )
3491 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
3492 end
3493
3494
3495 function manytimes (fsuffix)
3496 --
3497 -- Compile up to five times.
3498 -- fsuffix is "" for print, "_html" for HTML output
3499 --
3500 if onetime(fsuffix) == true then
3501 if onetime(fsuffix) == true then
3502 if onetime(fsuffix) == true then
3503 if onetime(fsuffix) == true then
```

```
3504 if onetime(fsuffix) == true then
3505 end end end end
3506 end
3507
3508
3509 function verifyfileexists (filename)
3510 --
3511 -- Exit if the given file does not exist.
3512 --
3513 if (lfs.attributes ( filename , "modification" ) == nil ) then
3514     print ("lwarpmk: ===")
3515     print ("lwarpmk: " .. filename .. " not found." );
3516     print ("lwarpmk: ===")
3517     os.exit (1) ;
3518 end
3519 end
3520
3521
3522
3523 function pdftohtml ()
3524 --
3525 -- Convert <project>_html.pdf into HTML files:
3526 --
3527 -- Convert to text:
3528 print ("lwarpmk: Converting " .. sourcename
3529     .. "_html.pdf to " .. sourcename .. "_html.html")
3530 os.execute("pdftotext -enc " .. pdftotextenc .. " -nopgbrk -layout "
3531     .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
3532 -- Split the result into individual HTML files:
3533 splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
3534 end
3535
3536
3537 function removeaux ()
3538 --
3539 -- Remove auxiliary files:
3540 -- All aux files are removed since there may be many bbl*.aux files.
3541 --
3542 os.execute ( rmname .. " *.aux " ..
3543     sourcename .. ".toc " .. sourcename .. "_html.toc " ..
3544     sourcename .. ".lof " .. sourcename .. "_html.lof " ..
3545     sourcename .. ".lot " .. sourcename .. "_html.lot " ..
3546     sourcename .. ".idx " .. sourcename .. "_html.idx " ..
3547     sourcename .. ".ind " .. sourcename .. "_html.ind " ..
3548     sourcename .. ".log " .. sourcename .. "_html.log " ..
3549     sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
3550     " *_html_inc.* "
3551 )
3552 end
3553
```

```
3554 function checkhtmlpdfexists ()
3555 --
3556 -- Error if the HTML document does not exist.
3557 -- The lateximages are drawn from the HTML PDF version of the document,
3558 -- so "lwarpmk html" must be done before "lwarpmk limages".
3559 --
3560 local htmlpdffile = io.open(sourcename .. "_html.pdf", "r")
3561 if ( htmlpdffile == nil ) then
3562     print ("")
3563     print ("lwarpmk: ===")
3564     print ("lwarpmk: The HTML version of the document does not exist.")
3565     print ("lwarpmk: Enter \"lwarpmk html\" to compile the HTML version.")
3566     print ("lwarpmk: ===")
3567     os.exit(1)
3568 end
3569 io.close (htmlpdffile)
3570 end -- checkhtmlpdfexists
3571
3572
3573 function warnlimages ()
3574 --
3575 -- Warning of a missing lateximages.txt file:
3576 --
3577     print ("lwarpmk: ===")
3578     print ("lwarpmk: \"lateximages.txt\" does not exist.")
3579     print ("lwarpmk: Your project does not use SVG math or other lateximages,")
3580     print ("lwarpmk: or the file has been deleted somehow.")
3581     print ("lwarpmk: Use \"lwarpmk html\" to recompile your project")
3582     print ("lwarpmk: and recreate \"lateximages.txt\".")
3583     print ("lwarpmk: If your project does not use SVG math or other lateximages,")
3584     print ("lwarpmk: then \"lateximages.txt\" will never exist, and")
3585     print ("lwarpmk: \"lwarpmk limages\" will not be necessary.")
3586     print ("lwarpmk: ===")
3587 end -- warnlimages
3588
3589
3590 function checklimages ()
3591 --
3592 -- Check lateximages.txt to see if need to recompile first.
3593 -- If any entry has a page number of zero, then there were incorrect images.
3594 --
3595 print ("lwarpmk: Checking for a valid lateximages.txt file.")
3596 local limagesfile = io.open("lateximages.txt", "r")
3597 if ( limagesfile == nil ) then
3598     warnlimages ()
3599     os.exit(1)
3600 end
3601 -- Track warning to recompile if find a page 0
3602 local pagezerowarning = false
3603 -- Scan lateximages.txt
```

```

3604 for line in limagesfile:lines() do
3605     -- lwimgpage is the page number in the PDF which has the image
3606     -- lwimghash is true if this filename is a hash
3607     -- lwimgname is the lateximage filename root to assign for the image
3608     i,j,lwimgpage,lwimghash,lwimgname = string.find (line,"|(.*)|(.*)|(.*)|")
3609     -- For each entry:
3610     if ( (i~=nil) ) then
3611         -- If the page number is 0, image references are incorrect
3612         -- and must recompile the source document:
3613         if ( lwimgpage == "0" ) then
3614             pagezerowarning = true
3615         end
3616     end -- if i~=nil
3617 end -- do
3618 if ( pagezerowarning ) then
3619     print ("")
3620     print ("lwarpmk: ===")
3621     print ("lwarpmk: The document must be recompiled before creating the lateximages.")
3622     print ("lwarpmk: Enter \"lwarpmk html\" again, then try \"lwarpmk limages\" again.")
3623     print ("lwarpmk: ===")
3624     os.exit(1) ;
3625 end -- pagezerowarning
3626 end -- checklimages
3627
3628
3629 function createuniximage ( lwimgfullname )
3630 --
3631 -- Create one lateximage for Unix / Linux / Mac OS.
3632 --
3633 executecheckerror (
3634     cmdgroupopenname ..
3635     "pdfseparate -f " .. lwimgpage .. " -l " .. lwimgpage .. " " ..
3636     sourcename .. "_html.pdf " ..
3637     "lateximages" .. dirslash .. "lateximagetemp-%d" .. ".pdf" ..
3638     seqname ..
3639     -- Crop the image:
3640     "pdfcrop -- hires lateximages" .. dirslash .. "lateximagetemp-" ..
3641     lwimgpage .. ".pdf " ..
3642     "lateximages" .. dirslash .. lwimgname .. ".pdf" ..
3643     seqname ..
3644     -- Convert the image to svg:
3645     "pdftocairo -svg -noshrink lateximages" .. dirslash .. lwimgname .. ".pdf " ..
3646     "lateximages" .. dirslash .. lwimgname .. ".svg" ..
3647     seqname ..
3648     -- Remove the temporary files:
3649     rmname .. " lateximages" .. dirslash .. lwimgname .. ".pdf" .. seqname ..
3650     rmname .. " lateximages" .. dirslash .. "lateximagetemp-" .. lwimgpage .. ".pdf" ..
3651     cmdgroupclosename .. " >/dev/null " .. bname
3652     ,
3653     "File error trying to convert " .. lwimgfullname

```

```

3654)
3655-- Every 32 images, wait for completion at below normal priority,
3656-- allowing other image tasks to catch up.
3657numimageprocesses = numimageprocesses + 1
3658if ( numimageprocesses > 32 ) then
3659    numimageprocesses = 0
3660    print ( "lwarpmk: waiting" )
3661    executecheckerror ( "wait" , "File error trying to wait.")
3662end
3663end -- createuniximage
3664
3665
3666function createwindowsimage ( lwimgfullname )
3667--
3668-- Create one lateximage for Windows.
3669--
3670-- Every 32 images, wait for completion at below normal priority,
3671-- allowing other image tasks to catch up.
3672numimageprocesses = numimageprocesses + 1
3673if ( numimageprocesses > 32 ) then
3674    numimageprocesses = 0
3675    thiswaitcommand = "/WAIT /BELOWNORMAL"
3676    print ( "lwarpmk: waiting" )
3677else
3678    thiswaitcommand = ""
3679end
3680-- Execute the image generation command
3681executecheckerror (
3682    "start /B " .. thiswaitcommand .. " \"\" lwarp_one_limage " ..
3683    lwimgpage .. " " ..
3684    lwimgghash .. " " ..
3685    lwimgname .. " " ..
3686    sourcename .. " <nul >nul"
3687    ,
3688    "File error trying to create image."
3689)
3690end -- createwindowsimage
3691
3692
3693function createonelateximage ( line )
3694--
3695-- Given the next line of lateximages.txt, convert a single image.
3696--
3697-- lwimgpage is the page number in the PDF which has the image
3698-- lwimgghash is true if this filename is a hash
3699-- lwimgname is the lateximage filename root to assign for the image
3700i,j,lwimgpage,lwimgghash,lwimgname = string.find (line,"|(.*)|(.*)|(.*)|")
3701-- For each entry:
3702if ( (i~=nil) ) then
3703    -- Skip if the page number is 0:

```

```

3704     if ( lwimgpage == "0" ) then
3705         pagezerowarning = true
3706     else
3707         -- Skip is this image is hashed and already exists:
3708         local lwimgfullname = "lateximages" .. dirsslash .. lwimgname .. ".svg"
3709         if (
3710             (lwimghash ~= "true") or
3711             (lfs.attributes(lwimgfullname,"mode")==nil) -- file not exists
3712         )
3713         then -- not hashed or not exists:
3714             -- Print the name of the file being generated:
3715             print ( "lwarpmk: " .. lwimgname )
3716             -- Touch/create the dest so that only once instance tries to build it:
3717             executecheckerror (
3718                 newtouchname .. " " .. lwimgfullname ,
3719                 "File error trying to touch " .. lwimgfullname
3720             )
3721             -- Separate out the image into its own single-page pdf:
3722             if opsystem=="Unix" then
3723                 createuniximage (lwimgfullname)
3724             elseif opsystem=="Windows" then
3725                 createwindowsimage (lwimgfullname)
3726             end
3727         end -- not hashed or not exists
3728     end -- not page 0
3729 end -- not nil
3730 end -- createonelateximage
3731
3732
3733 function createlateximages ()
3734 --
3735 -- Create lateximages based on lateximages.txt:
3736 --
3737 -- See if the document must be recompiled first:
3738 checklimages ()
3739 -- See if the print version exists:
3740 checkhtmlpdfexists ()
3741 -- Attempt to create the lateximages:
3742 print ("lwarpmk: Creating lateximages.")
3743 local limagesfile = io.open("lateximages.txt", "r")
3744 if ( limagesfile == nil ) then
3745     warnlateximages ()
3746     os.exit(1)
3747 end
3748 -- Create the lateximages directory, ignore error if already exists
3749 err = os.execute("mkdir lateximages")
3750 -- For Windows, create lwarp_one_limage.cmd from lwarp_one_limage.txt:
3751 if opsystem=="Windows" then
3752     executecheckerror (
3753         cpname .. " lwarp_one_limage.txt lwarp_one_limage.cmd" ,

```

```

3754         "File error trying to copy lwarp_one_limage.txt to lwarp_one_limage.cmd"
3755     )
3756 end -- create lwarp_one_limage.cmd
3757 -- Track the number of parallel processes
3758 numimageprocesses = 0
3759 -- Track warning to recompile if find a page 0
3760 pagezerowarning = false
3761 -- Scan lateximages.txt
3762 for line in limagesfile:lines() do
3763     createonelateximage ( line )
3764 end -- do
3765 io.close(limagesfile)
3766 print ( "lwarpmk limages: ===")
3767 print ( "lwarpmk limages: Wait a moment for the images to complete" )
3768 print ( "lwarpmk limages:   before reloading the page." )
3769 print ( "lwarpmk limages: ===")
3770 print ( "lwarpmk limages: Done." )
3771 if ( pagezerowarning == true ) then
3772     print ( "lwarpmk limages: WARNING: Images will be incorrect." )
3773     print ( "lwarpmk limages:   Enter \"lwarpmk cleanlimages\", then" )
3774     print ( "lwarpmk limages:   recompile the document one more time, then" )
3775     print ( "lwarpmk limages:   repeat \"lwarpmk images\" again." )
3776 end -- pagezerowarning
3777 end -- function
3778
3779
3780 function compilelatexmk ( fsuffix )
3781 --
3782 -- Use latexmk to compile source and index:
3783 -- fsuffix is "" for print, or "_html" for HTML
3784 --
3785 -- Maybe select the shell-escape option:
3786 local thisshellescape = " "
3787 if ( shellescape == "true" ) then
3788     thisshellescape = " -shell-escape "
3789 else
3790     thisshellescape = " "
3791 end
3792 -- The recorder option is required to detect changes in <project>.tex
3793 -- while we are loading <project>_html.tex.
3794 executecheckerror (
3795     "latexmk -pdf -dvi- -ps- -recorder "
3796     .. "-e "
3797     .. opquote
3798     .. "$makeindex = q/" -- $
3799     .. xindycmd
3800     .. " -M " .. xindystyle
3801     .. " -C " .. xindycodepage
3802     .. " -L " .. xindylanguage .. " /"
3803     .. opquote

```

```
3804    .. "-pdflatex=\"\" .. latexname .. thisshellescape .." %0 %S\" \"
3805    .. sourcename..fsuffix ..\".tex\"
3806    ,
3807    "Compile error.\"
3808)
3809end -- function
3810
3811
3812function convertpdftosvg ()
3813--
3814-- Converts PDF files to SVG files.
3815-- The filenames are arg[2] and up.
3816-- arg[1] is the command "pdftosvg\".
3817--
3818for i = 2 , #arg do
3819    if (lfs.attributes(arg[i],\"mode\")==nil) then
3820        print (\"lwarpmk: File \" .. arg[i] .. \"\" does not exist.\")
3821    else
3822        print (\"lwarpmk: Converting \" .. arg[i] .. \"\")
3823        os.execute ( \"pdftocairo -svg \" .. arg[i] )
3824    end -- if
3825end -- do
3826end --function
3827
3828
3829
3830-- Start of the main code: --
3831
3832
3833-- lwarpmk --version :
3834
3835if (arg[1] == \"--version\") then
3836print ( \"lwarpmk: \" .. printversion )
3837
3838else -- not --version
3839
3840
3841-- print intro:
3842
3843print (\"lwarpmk: \" .. printversion .. \" Automated make for the LaTeX lwarp package.\")
3844
3845
3846-- lwarpmk print:
3847
3848if arg[1] == \"print\" then
3849loadconf ()
3850if ( latexmk == \"true\" ) then
3851    compilelatexmk (\"\")
3852    print (\"lwarpmk: Done.\")
3853else -- not latexmk
```

```
3854     verifyfileexists (sourcename .. ".tex") ;
3855     -- See if up to date:
3856     if (
3857         ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
3858         (
3859             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3860             lfs.attributes ( sourcename .. ".pdf" , "modification" )
3861         )
3862     ) then
3863         -- Recompile if not yet up to date:
3864         manytimes("")
3865         print ("lwarpmk: Done.") ;
3866     else
3867         print ("lwarpmk: " .. sourcename .. ".pdf is up to date.") ;
3868     end
3869 end -- not latexmk
3870
3871
3872 -- lwarpmk print1:
3873
3874 elseif arg[1] == "print1" then
3875     loadconf ()
3876     verifyfileexists (sourcename .. ".tex") ;
3877     onetime("")
3878     print ("lwarpmk: Done.") ;
3879
3880
3881 -- lwarp printindex:
3882 -- Compile the index then touch the source
3883 -- to trigger a recompile of the document:
3884
3885 elseif arg[1] == "printindex" then
3886     loadconf ()
3887     print ("lwarpmk: Processing the index.")
3888     os.execute(
3889         xindycmd
3890         .. " -M " .. xindystyle
3891         .. " -C " .. xindycodepage
3892         .. " -L " .. xindylanguage
3893         .. " " .. sourcename .. ".idx")
3894     print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3895     refreshdate ()
3896     print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3897     print ("lwarpmk: Done.")
3898
3899
3900 -- lwarp printglossary:
3901 -- Compile the glossary then touch the source
3902 -- to trigger a recompile of the document:
3903
```

```
3904 elseif arg[1] == "printglossary" then
3905 loadconf ()
3906 print ("lwarpmk: Processing the glossary.")
3907
3908 os.execute(glossarycmd ..
3909     " -L " .. xindylanguage ..
3910     " -C " .. xindycodepage ..
3911     " -I xindy -M " .. sourcename ..
3912     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".glg "
3913     .. sourcename .. ".glo")
3914 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3915 refreshdate ()
3916 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3917 print ("lwarpmk: Done.")
3918
3919
3920 -- lwarpmk html:
3921
3922 elseif arg[1] == "html" then
3923 loadconf ()
3924 if ( latexmk == "true" ) then
3925     compilelatexmk ("_html")
3926     pdftohtml ()
3927     print ("lwarpmk: Done.")
3928 else -- not latexmk
3929     verifyfileexists ( sourcename .. ".tex" ) ;
3930     -- See if exists and is up to date:
3931     if (
3932         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3933         (
3934             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3935             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3936         )
3937     ) then
3938         -- Recompile if not yet up to date:
3939         manytimes("_html")
3940         pdftohtml ()
3941         print ("lwarpmk: Done.")
3942     else
3943         print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3944     end
3945 end -- not latexmk
3946
3947
3948 -- lwarpmk html1:
3949
3950 elseif arg[1] == "html1" then
3951     loadconf ()
3952     verifyfileexists ( sourcename .. ".tex" ) ;
3953     onetime("_html")
```

```
3954     pdftohtml ()
3955     print ("lwarpmk: Done.")
3956
3957
3958-- lwarpmk pdftohtml:
3959elseif arg[1] == "pdftohtml" then
3960     loadconf ()
3961     pdftohtml ()
3962
3963
3964-- lwarpmk htmlindex:
3965-- Compile the index then touch the source
3966-- to trigger a recompile of the document:
3967
3968elseif arg[1] == "htmlindex" then
3969loadconf ()
3970print ("lwarpmk: Processing the index.")
3971os.execute(
3972     xindycmd
3973     .. " -M " .. xindystyle
3974     .. " -L " .. xindylanguage
3975     .. " -C " .. xindycodepage
3976     .. " " .. sourcename .. "_html.idx"
3977)
3978print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3979refreshdate ()
3980print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3981print ("lwarpmk: Done.")
3982
3983
3984-- lwarpmk htmlglossary:
3985-- Compile the glossary then touch the source
3986-- to trigger a recompile of the document:
3987
3988elseif arg[1] == "htmlglossary" then
3989loadconf ()
3990print ("lwarpmk: Processing the glossary.")
3991
3992os.execute(glossarycmd ..
3993     " -L " .. xindylanguage ..
3994     " -C " .. xindycodepage ..
3995     " -I xindy -M " .. sourcename ..
3996     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3997     "_html.gls " .. sourcename .. "_html.glo")
3998
3999print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
4000refreshdate ()
4001print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
4002print ("lwarpmk: Done.")
4003
```

```
4004
4005-- lwarpmk limages:
4006-- Scan the lateximages.txt file to create lateximages.
4007
4008elseif arg[1] == "limages" then
4009loadconf ()
4010print ("lwarpmk: Processing images.")
4011createlateximages ()
4012print ("lwarpmk: Done.")
4013
4014
4015-- lwarpmk again:
4016-- Touch the source to trigger a recompile.
4017
4018elseif arg[1] == "again" then
4019loadconf ()
4020print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
4021refreshdate ()
4022print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
4023print ("lwarpmk: Done.")
4024
4025
4026-- lwarpmk clean:
4027-- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
4028
4029elseif arg[1] == "clean" then
4030loadconf ()
4031removeaux ()
4032print ("lwarpmk: Done.")
4033
4034
4035-- lwarpmk cleanall
4036-- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
4037-- and also project.pdf, *.html
4038
4039elseif arg[1] == "cleanall" then
4040loadconf ()
4041removeaux ()
4042os.execute ( rmname .. " " ..
4043    sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
4044    "*.html"
4045    )
4046print ("lwarpmk: Done.")
4047
4048
4049-- lwarpmk cleanlimages
4050-- Remove images from the lateximages directory.
4051
4052elseif arg[1] == "cleanlimages" then
4053loadconf ()
```

```

4054 os.execute ( rmname .. " lateximages/*" )
4055 print ("lwarpmk: Done.")
4056
4057 -- lwarpmk pdftosvg <list of file names>
4058 -- Convert PDf files to SVG using pdftocairo
4059 elseif arg[1] == "pdftosvg" then
4060 convertpdftosvg ()
4061 print ("lwarpmk: Done.")
4062
4063
4064 -- lwarpmk with no argument :
4065
4066 elseif (arg[1] == nil) then
4067 printhelp ()
4068
4069
4070 -- lwarpmk -h or lwarpmk --help :
4071
4072 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
4073 printusage ()
4074
4075
4076 -- Unknown command:
4077
4078 else
4079 printhelp ()
4080 print ("\nlwarpmk: ***** Unknown command \""..arg[1].."\". *****\n")
4081 end
4082
4083 end -- not --version
4084 \end{filecontents*}
4085 % \end{Verbatim}% for syntax highlighting

4086 \end{LWR@createlwarpmk}

```

38 Stacks

for HTML output: 4087 \begin{warpHTML}



Stacks are used to remember how to close sections and list items. Before a new section is started, previously nested sections and items must be closed out (un-nested) in proper order. Note that starting a new section may close several levels of previously nested items at the same time. For example, starting a new `\section` would close any currently open subsection, subsubsection, and paragraph. General environments are not nested on the stack since they have their own close mechanism. List environments are nested, and items inside those environments are nested one level

deeper still. List environments may be nested inside other list environments, and list items are nested inside list environments as well. Thus, the stack may have items which are not necessarily in order, since a description may contain an enumerate, for example. Depths to be recorded in `\LWR@closedepthone`, etc.

38.1 Assigning depths

initial depths for empty stack entries:

```
4088 \newcommand*{\LWR@depthnone}{-5}
```

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

```
4089 \newcommand*{\LWR@depthfinished}{-4}
4090 \newcommand*{\LWR@depthpart}{-1}
4091 \newcommand*{\LWR@depthchapter}{0}
4092 \newcommand*{\LWR@depthsection}{1}
4093 \newcommand*{\LWR@depthsubsection}{2}
4094 \newcommand*{\LWR@depthsubsubsection}{3}
4095 \newcommand*{\LWR@depthparagraph}{4}
4096 \newcommand*{\LWR@depthsubparagraph}{5}
```

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

```
4097 \newcommand*{\LWR@depthlist}{6}
```

used by `\item`:

```
4098 \newcommand*{\LWR@depthlistitem}{7}
```

38.2 Closing actions

A stack to record the action to take to close each nesting level: Add more levels of stack if necessary for a very deeply nested document, adding to `\pushclose` and `\popclose` as well.

```
4099 \newcommand*{\LWR@closeone}{}% top of the stack
4100 \newcommand*{\LWR@closetwo}{}
4101 \newcommand*{\LWR@closethree}{}
4102 \newcommand*{\LWR@closefour}{}
4103 \newcommand*{\LWR@closefive}{}
4104 \newcommand*{\LWR@closesix}{}
4105 \newcommand*{\LWR@closeseven}{}
4106 \newcommand*{\LWR@closeeight}{}%
```

```

4107 \newcommand*\LWR@closenine{}\}
4108 \newcommand*\LWR@closeten{}\}
4109 \newcommand*\LWR@closeeleven{}\}
4110 \newcommand*\LWR@closetwelve{}\}

```

38.3 Closing depths

A stack to record the depth of each level:



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

Ex:

```

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

```

```

4111 \newcommand*\LWR@closedepthone{}\LWR@depthnone}% top of the stack
4112 \newcommand*\LWR@closedepthtwo{}\LWR@depthnone}
4113 \newcommand*\LWR@closedepththree{}\LWR@depthnone}
4114 \newcommand*\LWR@closedepthfour{}\LWR@depthnone}
4115 \newcommand*\LWR@closedepthfive{}\LWR@depthnone}
4116 \newcommand*\LWR@closedepthsix{}\LWR@depthnone}
4117 \newcommand*\LWR@closedepthseven{}\LWR@depthnone}
4118 \newcommand*\LWR@closedeptheight{}\LWR@depthnone}
4119 \newcommand*\LWR@closedepthnine{}\LWR@depthnone}
4120 \newcommand*\LWR@closedephten{}\LWR@depthnone}
4121 \newcommand*\LWR@closedeptheleven{}\LWR@depthnone}
4122 \newcommand*\LWR@closedephtwelve{}\LWR@depthnone}

```

38.4 Pushing and popping the stack

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

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

```

4123 \NewDocumentCommand{\pushclose}{m m}
4124 {
4125 \global\let\LWR@closetwelve\LWR@closeeleven

```

```

4126 \global\let\LWR@closeeleven\LWR@closeten
4127 \global\let\LWR@closeten\LWR@closenine
4128 \global\let\LWR@closenine\LWR@closeeight
4129 \global\let\LWR@closeeight\LWR@closeseven
4130 \global\let\LWR@closeseven\LWR@closesix
4131 \global\let\LWR@closesix\LWR@closefive
4132 \global\let\LWR@closefive\LWR@closefour
4133 \global\let\LWR@closefour\LWR@closethree
4134 \global\let\LWR@closethree\LWR@closetwo
4135 \global\let\LWR@closetwo\LWR@closeone
4136 \global\let\LWR@closeone#1
4137 \global\let\LWR@closedepthtwelve\LWR@closedeptheleven
4138 \global\let\LWR@closedeptheleven\LWR@closedephten
4139 \global\let\LWR@closedephten\LWR@closedepthnine
4140 \global\let\LWR@closedepthnine\LWR@closedeptheight
4141 \global\let\LWR@closedeptheight\LWR@closedepthseven
4142 \global\let\LWR@closedepthseven\LWR@closedepthsix
4143 \global\let\LWR@closedepthsix\LWR@closedepthfive
4144 \global\let\LWR@closedepthfive\LWR@closedepthfour
4145 \global\let\LWR@closedepthfour\LWR@closedepththree
4146 \global\let\LWR@closedepththree\LWR@closedepthtwo
4147 \global\let\LWR@closedepthtwo\LWR@closedepthone
4148 \global\let\LWR@closedepthone#2
4149 }

```

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

```

4150 \newcommand*{\popclose}
4151 {
4152 \global\let\LWR@closeone\LWR@closetwo
4153 \global\let\LWR@closetwo\LWR@closethree
4154 \global\let\LWR@closethree\LWR@closefour
4155 \global\let\LWR@closefour\LWR@closefive
4156 \global\let\LWR@closefive\LWR@closesix
4157 \global\let\LWR@closesix\LWR@closeseven
4158 \global\let\LWR@closeseven\LWR@closeeight
4159 \global\let\LWR@closeeight\LWR@closenine
4160 \global\let\LWR@closenine\LWR@closeten
4161 \global\let\LWR@closeten\LWR@closeeleven
4162 \global\let\LWR@closeeleven\LWR@closetwelve
4163 \global\let\LWR@closedepthone\LWR@closedepthtwo
4164 \global\let\LWR@closedepthtwo\LWR@closedepththree
4165 \global\let\LWR@closedepththree\LWR@closedepthfour
4166 \global\let\LWR@closedepthfour\LWR@closedepthfive
4167 \global\let\LWR@closedepthfive\LWR@closedepthsix
4168 \global\let\LWR@closedepthsix\LWR@closedepthseven
4169 \global\let\LWR@closedepthseven\LWR@closedeptheight
4170 \global\let\LWR@closedeptheight\LWR@closedepthnine
4171 \global\let\LWR@closedepthnine\LWR@closedephten

```

```

4172 \global\let\LWR@closedepthten\LWR@closedeptheleven
4173 \global\let\LWR@closedeptheleven\LWR@closedepthtwelve
4174 }

4175 \end{warpHTML}

```

39 Data arrays

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

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

To assign an empty value:

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

for HTML output: 4176 \begin{warpHTML}

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

```

4177 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
4178 \ifstrempy{#3}%
4179 {\csdef{#1#2}{}}%
4180 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
4181 }

```

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

```


4182 \newcommand*{\LWR@getexparray}[2]{\csuse{#1#2}}

4183 \end{warpHTML}

```

40 Localizing catcodes

for HTML & PRINT: 4184 \begin{warppall}

 **misplaced alignment tab character &** Place `\StartDefiningTabulars` and `\StopDefiningTabulars` before and after defining macros or environments which include the tabular & character in their definitions.

The catcode of & must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

```
misplaced alignment tab character &
```

`\StartDefiningTabulars` Place before defining something with & in it.

```
4185 \newcommand{\StartDefiningTabulars}{%
4186 \LWR@traceinfo{StartDefiningTabulars}%
4187 \warpHTMLonly{\catcode'\&=\active}%
4188 }
```

`\StopDefiningTabulars` Place after defining something with & in it.

```
4189 \newcommand{\StopDefiningTabulars}{%
4190 \LWR@traceinfo{StopDefiningTabulars}%
4191 \warpHTMLonly{\catcode'\&=4}%
4192 }
```

`Bool LWR@mathmacro` True if currently defining math macros. Used to disable svg math hashing and MathJax math contents while defining a macro using inline math. Begin a macro, it is not guaranteed that the contents are static, and so the image must be unique. The contents also almost certainly will not be parsed correctly by MathJax.

```
4193 \newbool{LWR@mathmacro}
4194 \boolfalse{LWR@mathmacro}
```

`\StartDefiningMath` Place before defining something with \$ in it.

```
4195 \newcommand{\StartDefiningMath}{%
4196 \LWR@traceinfo{StartDefiningMath}%
4197 \warpHTMLonly{\catcode'\$=\active}%
4198 }
```

`\StopDefiningMath` Place after defining something with \$ in it.

```
4199 \newcommand{\StopDefiningMath}{%
4200 \LWR@traceinfo{StopDefiningMath}%
4201 \warpHTMLonly{\catcode'\$=3}% math shift
4202 }

4203 \end{warpall}
```

41 Localizing dynamic math

Inline svg math usually uses a hash of its contents to generate lateximages which are reusable for multiple instances with the same contents. If the contents may change for each use, such as depending on the current value of a counter, then `\StartDynamicMath` must be used before the inline math expression, and `\StopDynamicMath` must be used after.

For MathJax, the inline math expression is usually printed for MathJax to interpret. When marked as dynamic math, the following inline math expression will be displayed as an unhashed inline svg image instead.

For existing code and packages, it may be possible to patch macros after they have been defined, using the **xpatch** package, which is pre-loaded by **lwarp**:

```
\xpatchcmd{\macroname}
  {$math expression$}
  {\StartDynamicMath$math expression$\StopDynamicMath}
  {}
  {\typeout{Error patching macroname.}}
```

for HTML & PRINT: 4204 `\begin{warpall}`

Bool LWR@dynamicmath True to mark inline math which is dynamic in nature, thus should not be hashed for reuse.
Default: false

```
4205 \newbool{LWR@dynamicmath}
4206 \boolfalse{LWR@dynamicmath}
```

`\StartDynamicMath` Place before using `$... $` or `\(... \)` if the contents of the math are not static, depending on counters or dynamic macros.

```
4207 \newcommand{\StartDynamicMath}{%
4208 \LWR@traceinfo{StartDynamicMath}%
4209 \booltrue{LWR@dynamicmath}%
4210 }
```

`\StopDynamicMath` Place after using `$... $` or `\(... \)` with dynamic contents.

```
4211 \newcommand{\StopDynamicMath}{%
4212 \LWR@traceinfo{StopDynamicMath}%
4213 \boolfalse{LWR@dynamicmath}%
4214 }
```

```
4215 \end{warpall}
```

42 Sanitizing labels and filenames

Special handling for underscores in labels and filenames.

for HTML output: 4216 \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.

```
4217 \newcommand*{\LWR@sanitized}{}%
```

`\LWR@sanitize` $\{ \langle text \rangle \}$

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

```
4218 \newcommand*{\LWR@sanitize}[1]{%
4219 \LWR@traceinfo{\LWR@sanitize: !#1!}%
4220 \edef\LWR@sanitized{#1}%
4221 \LWR@traceinfo{\LWR@sanitize expanded: !\LWR@sanitized!}%
4222 \edef\LWR@sanitized{\detokenize\expandafter{\LWR@sanitized}}%
4223 \LWR@traceinfo{\LWR@sanitize result: !\LWR@sanitized!}%
4224 }
```

```
4225 \end{warpHTML}
```

43 HTML entities

for HTML output: 4226 \begin{warpHTML}

HTML entites and HTML Unicode entities:

```
4227 \let\LWR@origampersand\&
```

`\HTMLentity` $\{ \langle entitytag \rangle \}$

```
4228 \newcommand*{\HTMLentity}[1]{%
4229 % \LWR@traceinfo{\HTMLentity \detokenize{#1}}%
4230 \begingroup%
4231 \LWR@FBcancel%
```

```

4232 \LWR@origampersand#1;%
4233 \endgroup
4234 % \LWR@traceinfo{HTMLentity done}%
4235 }

```

`\HTMLunicode` `{\hex_unicode}`

```

4236 \newcommand*{\HTMLunicode}[1]{\HTMLentity{\LWR@origpound{x#1}}

```

`\&`

```

4237 \renewrobustcmd*{\&}{\HTMLentity{amp}}

```

`\textless`

`\textgreater`

```

4238 \let\LWR@origtextless\textless
4239 \renewcommand*{\textless}{\HTMLentity{lt}}
4240
4241 \let\LWR@origtextgreater\textgreater
4242 \renewcommand*{\textgreater}{\HTMLentity{gt}}

4243 \end{warpHTML}

```

44 HTML filename generation

The filename of the homepage is set to `\HomeHTMLFilename.html`. The filenames of additional sections start with `\HTMLFilename`, to which is appended a section number or a simplified section name, depending on `FileSectionNames`.

for HTML & PRINT: 4244 `\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.

```

4245 \providecommand*{\BaseJobname}{\jobname}

```

`\HTMLFilename` The prefix for all generated HTML files other than the home page, defaulting to empty. See section [8.4.1](#).

```

4246 \providecommand*{\HTMLFilename}{}

```

`\HomeHTMLFilename` The filename of the home page, defaulting to the `\BaseJobname`. See section 8.4.1.

```
4247 \providecommand*{\HomeHTMLFilename}{\BaseJobname}
```

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

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

```
4248 \newcommand*{\SetHTMLFileNumber}[1]{%
4249 \setcounter{LWR@htmlfilenumber}{#1}%
4250 }
```

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

Defaults to use section names in the filenames.

```
4251 \newbool{FileSectionNames}
4252 \booltrue{FileSectionNames}
```

```
4253 \end{warppall}
```

for HTML output: `\begin{warppHTML}`

```
4254 \begin{warppHTML}
```

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

```
4255 \newcounter{LWR@htmlfilenumber}
4256 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
4257 \newcommand*{\LWR@htmlsectionfilename}[1]{%
4258 \LWR@traceinfo{LWR@htmlsectionfilename A !\detokenize{#1}!}%
```

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

```
4259 % \LWR@traceinfo{about to assign temp}%
4260 \edef\LWR@tempone{#1}%
4261 \LWR@traceinfo{about to compare with ??}%
4262 \ifthenelse{\equal{\LWR@tempone}{??}}%
```

```

4263 {\LWR@traceinfo{found ??}}%
4264 {\LWR@traceinfo{not found ??}}%
4265 \LWR@traceinfo{about to compare with zero or empty}%
4266 \ifthenelse{%
4267     \equal{\LWR@tempone}{0}%
4268     \OR \equal{\LWR@tempone}{}%
4269     \OR \equal{\LWR@tempone}{??}%
4270 }%
4271 {%
4272     \LWR@traceinfo{\LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
4273     \HomeHTMLFilename.html%
4274 }%

```

For a H₁E_X section named “Index” or “index” without a prefix, create a filename with a leading underscore to avoid colliding with the HTML filename index.html:

```

4275 {%
4276     \LWR@traceinfo{\LWR@htmlsectionfilename C \LWR@tempone}%
4277     \ifthenelse{%
4278         \equal{\HTMLFilename}{%} \AND
4279         \equal{\LWR@tempone}{Index} \OR
4280         \equal{\LWR@tempone}{index}%
4281     }%
4282     {%
4283         \LWR@traceinfo{Prefixing the index name with an underscore.}%
4284         \_#1.html%
4285     }%

```

Otherwise, create a filename with the chosen prefix:

```

4286     {\HTMLFilename#1.html}%
4287 }%
4288 \LWR@traceinfo{\LWR@htmlsectionfilename Z}%
4289 }

```

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

Prints the filename for the given label

```

4290 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
4291 \LWR@traceinfo{\LWR@htmlrefsectionfilename: !\detokenize{#1}!}%

```

`\LWR@nullfonts` to allow math in a section name.

```

4292 \begingroup%
4293 \LWR@nullfonts%
4294 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
4295 \endgroup%

```

```

4296 \LWR@traceinfo{\LWR@htmlrefsectionfilename: done}%
4297 }

4298 \end{warpHTML}

```

45 Homepage link

for HTML output: 4299 \begin{warpHTML}

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

```

4300 \newcommand*{\LinkHome}{%
4301 \LWR@subhyperrefclass{\HomeHTMLFilename.html}{Home}{linkhome}%
4302 }

```

`\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.

```

4303 \newcommand*{\LWR@topnavigation}{
4304 \LWR@htmlelementclassline{nav}{topnavigation}{\LinkHome}
4305 }

```

`\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.

```

4306 \newcommand*{\LWR@botnavigation}{
4307 \LWR@htmlelementclassline{nav}{botnavigation}{\LinkHome}
4308 }

4309 \end{warpHTML}

```

46 \LWRPrintStack diagnostic tool



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

for HTML output: 4310 \begin{warpHTML}

`\LWRPrintStack` Prints the closedepth stack.

```

4311 \newcommand*{\LWR@subprintstack}{
4312 \LWR@closedepthone\ \LWR@closedepthtwo\ \LWR@closedepththree\
4313 \LWR@closedepthfour\ \LWR@closedepthfive\ \LWR@closedepthsix\
4314 \LWR@closedepthseven\ \LWR@closedeptheight\ \LWR@closedepthnine\
4315 \LWR@closedephten\ \LWR@closedeptheleven\ \LWR@closedephtwelve\
4316 }
4317
4318 \newcommand*{\LWRPrintStack}{
4319 \LWR@startpars
4320 \LWR@subprintstack
4321 }

4322 \end{warpHTML}

```

for PRINT output: 4323 `\begin{warpprint}`

```

4324 \newcommand*{\LWRPrintStack}{}

4325 \end{warpprint}

```

47 Closing stack levels

for HTML output: 4326 `\begin{warpHTML}`

Close one nested level:

```

4327 \newcommand*{\LWR@closeoneprevious}{\%
4328
4329 \LWR@closeone
4330
4331 \popclose
4332 }

```

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

```

4333 \newcommand*{\LWR@closeprevious}[1]{
4334 \LWR@traceinfo{\LWR@closeprevious to depth #1, depths are \LWR@subprintstack}%

```

Close any pending paragraph:

```

4335 \LWR@stoppars%

```

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

```

4336 \whileboolexpr{test{\ifnumcomp{\LWR@closedepthone}{>}{#1}}}%
4337 {%
4338   \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
4339   \LWR@closeoneprevious%
4340 }%
4341 \ifboolexpr{test{\ifnumcomp{\LWR@closedepthone}{=}{#1}}}%
4342 {%
4343   \LWR@traceinfo{\LWR@closeprevious: closing out depth \LWR@closedepthone}%
4344   \LWR@closeoneprevious%
4345 }{}%
4346 \LWR@traceinfo{\LWR@closeprevious: done, depths are \LWR@subprintstack}%
4347 }

4348 \end{warpHTML}

```

48 PDF pages and styles

for HTML output: 4349 \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.

```

4350 \newcommand{\LWR@forcenewpage}{%
4351 \LWR@traceinfo{\LWR@forcenewpage}%
4352 \ifinner\else%
4353 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
4354 \fi%
4355 }

```

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

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

```

4356 \renewcommand*{\pagestyle}[1]{}

```

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

```

4357 \renewcommand*{\thispagestyle}[1]{}

```

```

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

4358 \renewcommand*{\markboth}[2]{\}

\markright {\langle right\rangle}

4359 \renewcommand*{\markright}[1]{\}

\raggedbottom

4360 \renewcommand*{\raggedbottom}{\}

\flushbottom

4361 \renewcommand*{\flushbottom}{\}

\sloppy

4362 \renewcommand*{\sloppy}{\}

\fussy

4363 \renewcommand*{\fussy}{\}

\pagenumbering * {\langle commands\rangle}

4364 \RenewDocumentCommand{\pagenumbering}{s m}{\}

4365 \end{warpHTML}

```

49 HTML tags, spans, divs, elements

for HTML output: 4366 \begin{warpHTML}

49.1 Mapping \TeX Sections to HTML Sections

```

4367 \newcommand*{\LWR@tagtitle}{h1}
4368 \newcommand*{\LWR@tagtitleend}{/h1}
4369 \newcommand*{\LWR@tagpart}{h2}

```

```

4370 \newcommand*{\LWR@tagpartend}{/h2}
4371 \newcommand*{\LWR@tagchapter}{h3}
4372 \newcommand*{\LWR@tagchapterend}{/h3}
4373 \newcommand*{\LWR@tagsection}{h4}
4374 \newcommand*{\LWR@tagsectionend}{/h4}
4375 \newcommand*{\LWR@tagsubsection}{h5}
4376 \newcommand*{\LWR@tagsubsectionend}{/h5}
4377 \newcommand*{\LWR@tagsubsubsection}{h6}
4378 \newcommand*{\LWR@tagsubsubsectionend}{/h6}
4379 \newcommand*{\LWR@tagparagraph}{span class="paragraph"}
4380 \newcommand*{\LWR@tagparagraphend}{/span}
4381 \newcommand*{\LWR@tagsubparagraph}{span class="subparagraph"}
4382 \newcommand*{\LWR@tagsubparagraphend}{/span}
4383
4384 \newcommand*{\LWR@tagregularparagraph}{p}

```

49.2 Babel-French tag modifications

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

(Emulates or patches code by DANIEL FLIPO.)

```

4385 \providecommand*{\LWR@FBcancel}{%}
4386
4387 \AtBeginDocument{%
4388 \ifundefined{frenchbsetup}%
4389 {}%
4390 {%
4391     \frenchbsetup{FrenchFootnotes=false}%
4392 %
4393     \LetLtxMacro\LWR@FBcancel\NoAutoSpacing%
4394     \renewrobustcmd*{\FBcolonspace}{%
4395         \begingroup%
4396         \LWR@FBcancel%
4397         \LWR@origampersand{}nbsp;%
4398         \endgroup%
4399     }%
4400     \renewrobustcmd*{\FBthinspace}{%
4401         \begingroup%
4402         \LWR@FBcancel%
4403         \LWR@origampersand\LWR@origpound{x202f;% \,
4404         \endgroup%
4405     }%
4406     \renewrobustcmd*{\FBguillspace}{%
4407         \begingroup%
4408         \LWR@FBcancel%
4409         \LWR@origampersand{}nbsp;% ~, for \og xyz \fg{

```

```

4410      \endgroup%
4411    }%
4412    \DeclareDocumentCommand{\FBmedkern}{-}{%
4413      \begingroup%
4414      \LWR@FBcancel%
4415      \LWR@origampersand\LWR@origpound{x202f;% \,
4416      \endgroup%
4417    }%
4418    \DeclareDocumentCommand{\FBthickkern}{-}{%
4419      \begingroup%
4420      \LWR@FBcancel%
4421      \LWR@origampersand{\nbsp;% ~
4422      \endgroup%
4423    }%
4424    \renewrobustcmd*{~}{\HTMLentity{nbsp}}% was overwritten by babel-french
4425    \ifFBunicode%
4426    \else%
4427      \DeclareTextSymbol{\FBtextellipsis}{LY1}{133}%
4428      \DeclareTextCommandDefault{\FBtextellipsis}{\textellipsis\xspace}%
4429    \fi%
4430 }%
4431 }

```

49.3 HTML tags

`\LWR@htmltagc` `{\tag}` Break ligatures and use upright apostrophes in HTML tags.

`\protect` is in case the tag appears in TOC, LOE, LOT.

```

4432 \newcommand*{\LWR@htmltagc}[1]{%
4433 \LWR@traceinfo{\LWR@htmltagc !\detokenize{#1}!}%
4434 \begingroup%
4435 \LWR@FBcancel%
4436 \ifmmode\else\protect\LWR@origttfamily\fi%
4437 \protect\LWR@origtextless%
4438 #1%
4439 \protect\LWR@origtextgreater%
4440 \endgroup%
4441 % \LWR@traceinfo{\LWR@htmltagc: done}%
4442 }

```

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.

```

4443 \newcommand*{\LWR@nestspanitem}{%
4444 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
4445 \LWR@origitem%
4446 }
4447
4448 \newenvironment*{\LWR@nestspan}
4449 {%
4450 \LWR@traceinfo{\LWR@nestspan starting}%
4451 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}%
4452 {%
4453   \LWR@traceinfo{\LWR@nestspan: inside a lateximage}%
4454 }%
4455 {% not in a lateximage
4456   \LWR@traceinfo{\LWR@nestspan: NOT inside a lateximage}%
4457   \addtocounter{\LWR@spandepth}{1}%
4458   \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
4459   \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
4460   \renewcommand{\BlockClassSingle}[2]{##2}%
4461   \renewcommand{\LWR@forcenewpage}{}%
4462   \renewcommand{\LWR@liststart}{%
4463     \let\item\LWR@nestspanitem%
4464   }%
4465   \renewcommand{\LWR@listend}{\LWR@htmltagc{br /}\LWR@htmltagc{br /}}%
4466 }% not in a lateximage
4467 \LWR@traceinfo{\LWR@nestspan starting: done}%
4468 }% starting env
4469 {% ending env
4470 \LWR@traceinfo{\LWR@nestspan ending}%
4471 \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}%
4472 }%
4473 {\addtocounter{\LWR@spandepth}{-1}}%
4474 \LWR@traceinfo{\LWR@nestspan ending: done}%
4475 }
4476
4477 \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.

```

4478 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
4479 \LWR@ensuredoingaparp%
4480 \LWR@htmltagc{#1}%
4481 \begin{\LWR@nestspan}%
4482 #2%
4483 \LWR@htmltagc{/#1}%
4484 \end{\LWR@nestspan}%

```

4485 }

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

```
4486 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
4487 \LWR@traceinfo{\LWR@htmlspanclass |#1|#2|}%
4488 \LWR@ensuredoingapar%
4489 \LWR@subhtml elementclass{span}{#1}{#2}%
4490 \begin{\LWR@nestspan}%
4491 #3%
4492 \LWR@htmltagc{/span}%
4493 \LWR@traceinfo{\LWR@htmlspanclass done}%
4494 \end{\LWR@nestspan}%
4495 }
```

\LWR@htmltag {*<tag>*}

Print an HTML tag: <tag>

```
4496 \newcommand*{\LWR@htmltag}[1]{%
4497 % \LWR@traceinfo{\LWR@htmltagb !\detokenize{#1}!}%
4498 \LWR@htmltagc{#1}%
4499 % \LWR@traceinfo{\LWR@htmltagb: done}%
4500 }
```

49.4 Block tags and comments

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

\LWR@htmlopencomment

\LWR@htmlclosecomment

```
4501 \newcommand*{\LWR@htmlopencomment}{%
4502 {%
4503 % \LWR@traceinfo{\LWR@htmlopencomment}%
4504 \begingroup%
4505 \LWR@FBcancel%
4506 \ifmode\else\protect\LWR@origttfamily\fi%
4507 \LWR@print@mbx{\LWR@origtextless{}!{-}{-}}%
4508 \endgroup%
4509 }%
4510 }
4511
4512 \newcommand*{\LWR@htmlclosecomment}{%
```

```

4513 {%
4514 % \LWR@traceinfo{\LWR@htmlclosecomment}%
4515 \begingroup%
4516 \LWR@FBcancel%
4517 \ifmmode\else\protect\LWR@origttfamily\fi%
4518 \LWR@print@mbox{{-}{-}\LWR@origtextgreater}%
4519 \endgroup%
4520 }%
4521 }

```

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

```

4522 \newcommand{\LWR@htmlcomment}[1]{%
4523 \LWR@htmlopencomment{}}%
4524 {%
4525 \LWR@origttfamily% break ligatures
4526 #1%
4527 }%
4528 \LWR@htmlclosecomment{}}

```

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

```

4529 \newcommand{\LWR@htmlblockcomment}[1]
4530 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars}

```

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

```

4531 \newcommand*{\LWR@htmlblocktag}[1]{%
4532 \LWR@stoppars%
4533 \LWR@htmltag{#1}%
4534 \LWR@startpars%
4535 }

```

49.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```

4536 \NewDocumentCommand{\LWR@subhtmlclass}{m O{} m}{%
4537 \LWR@traceinfo{\LWR@subhtmlclass !#1!#2!#3!}%
4538 \ifblank{#2}%

```

```

4539 {\LWR@htmltag{#1 class="#3"}}% empty option
4540 {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
4541 \LWR@traceinfo{\LWR@subhtmlmlementclass done}%
4542 }

```

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

```

4543 \NewDocumentCommand{\LWR@htmlmlementclass}{m o m}{%
4544 \LWR@stoppars%
4545 \LWR@subhtmlmlementclass{#1}[#2]{#3}%
4546 \LWR@startpars%
4547 }

```

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

```

4548 \newcommand*{\LWR@htmlmlementclassend}[2]{%
4549 \LWR@stoppars%
4550 \LWR@htmltag{/#1}%
4551 \ifbool{HTMLDebugComments}{%
4552   \LWR@htmlcomment{End of #1 ‘#2’}%
4553 }{}%
4554 \LWR@startpars%
4555 }

```

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

```

4556 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
4557 \LWR@htmlmlementclass{div}[#1]{#2}%
4558 }

```

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

```

4559 \newcommand*{\LWR@htmldivclassend}[1]{%
4560 \LWR@htmlmlementclassend{div}{#1}%
4561 }

```

49.6 Single-line elements

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

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

```

4562 \NewDocumentCommand{\LWR@htmlelementclassline}{m o m +m}{%
4563 \LWR@stoppars
4564 \LWR@subhtmlelementclass{#1}[#2]{#3}%
4565 #4%
4566 \LWR@htmltag{/#1}
4567 \LWR@startpars
4568 }

```

49.7 HTML5 semantic elements

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

```

4569 \newcommand*{\LWR@htmlelement}[1]{%
4570 \LWR@htmlblocktag{#1}
4571 }

```

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

```

4572 \newcommand*{\LWR@htmlelementend}[1]{%
4573 \LWR@stoppars
4574 \LWR@htmltag{/#1}
4575 \LWR@startpars
4576 }
4577
4578 \end{warpHTML}

```

49.8 High-level block and inline classes

These are high-level commands which allow the creation of arbitrary block or inline sections which may be formatted with css.

Nullified versions are provided for print mode.

For other direct-formatting commands, see section [86](#).

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

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

for PRINT output: `4579 \begin{warpprint}`
`4580 \NewDocumentEnvironment{BlockClass}{o m}{\}{\}%`
`4581 \end{warpprint}`

for HTML output: 4582 \begin{warpHTML}
 4583 \NewDocumentEnvironment{LWR@print@BlockClass}{o m}{-}{-}%
 4584 \NewDocumentEnvironment{LWR@HTML@BlockClass}{o m}%
 4585 {%
 4586 \LWR@origpar%
 4587 \LWR@htmldivclass[#1]{#2}%
 4588 }
 4589 {\LWR@htmldivclassend{#2}}
 4590
 4591 \LWR@formattedenv{BlockClass}
 4592 \end{warpHTML}

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

for HTML & PRINT: 4593 \begin{warpall}
 4594 \newcommand{\BlockClassSingle}[2]{#2}
 4595 \end{warpall}

for HTML output: 4596 \begin{warpHTML}
 4597 \newcommand{\LWR@HTML@BlockClassSingle}[2]{%
 4598 \LWR@origpar%
 4599 \LWR@htmlclassline{div}{#1}{#2}%
 4600 }
 4601
 4602 \LWR@formatted{BlockClassSingle}
 4603 \end{warpHTML}

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

for PRINT output: 4604 \begin{warpprint}
 4605 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
 4606 \end{warpprint}

for HTML output: 4607 \begin{warpHTML}
 4608 \NewDocumentCommand{\LWR@print@InlineClass}{o m +m}{#3}%
 4609
 4610 \NewDocumentCommand{\LWR@HTML@InlineClass}{o m +m}{%
 4611 \LWR@htmlspanclass[#1]{#2}{#3}%
 4612 }
 4613
 4614 \LWR@formatted{InlineClass}
 4615 \end{warpHTML}

Env LWR@BlockClassWP {<WPstyle>} {<HTMLstyle>} {<class>} Low-level interface for <div> classes with an automatic float ID. These are often used when \ifbool{FormatWP}.

for PRINT output: 4616 \begin{warpprint}
 4617 \NewDocumentEnvironment{LWR@BlockClassWP}{m m m}{-}{-}%

```
4618 \end{warpprint}
```

for HTML output:

```

4619 \begin{warpHTML}
4620 \NewDocumentEnvironment{LWR@print@LWR@BlockClassWP}{m m m}{-}{-}%
4621 \NewDocumentEnvironment{LWR@HTML@LWR@BlockClassWP}{m m m}%
4622 {%
4623 \LWR@stoppars%
4624 \ifbool{FormatWP}%
4625 {%
4626 \addtocounter{LWR@thisautoidWP}{1}%
4627 \LWR@htmltag{%
4628 div class="#3" %
4629 id="\LWR@print@mbbox{autoidWP-\arabic{LWR@thisautoidWP}}"%
4630 \ifblank{#1}{-}{ style="#1"}%
4631 }%
4632 }% FormatWP
4633 {% not FormatWP
4634 \LWR@htmltag{%
4635 div class="#3"%
4636 \ifblank{#2}{-}{ style="#2"}%
4637 }%
4638 }% not FormatWP
4639 \LWR@startpars%
4640 }
4641 {\LWR@htmldivclassend{#3}}
4642
4643 \LWR@formattedenv{LWR@BlockClassWP}
4644 \end{warpHTML}

```

49.9 Closing HTML tags

for HTML output:

```
4645 \begin{warpHTML}
```

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

```

4646 \newcommand*{\LWR@printclosepart}
4647 {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
4648 \newcommand*{\LWR@printclosechapter}
4649 {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
4650 \newcommand*{\LWR@printclosesection}
4651 {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
4652 \newcommand*{\LWR@printclosesubsection}
4653 {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
4654 \newcommand*{\LWR@printclosesubsubsection}
4655 {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
4656 \newcommand*{\LWR@printcloseparagraph}

```

```

4657     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
4658 \newcommand*{\LWR@printclosesubparagraph}
4659     {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}

```

Lists require closing HTML tags:

```

4660 \newcommand*{\LWR@printcloselistitem}
4661     {\LWR@htmltag{/li}}
4662 \newcommand*{\LWR@printclosedescitem}
4663     {\LWR@htmltag{/dd}}
4664 \newcommand*{\LWR@printcloseitemize}
4665     {\LWR@htmltag{/ul}}
4666 \newcommand*{\LWR@printcloseenumerate}
4667     {\LWR@htmltag{/ol}}
4668 \newcommand*{\LWR@printclosedescription}
4669     {\LWR@htmltag{/dl}}

```

```

4670 \end{warpHTML}

```

50 Paragraph handling

These commands generate the HTML paragraph tags when allowed and required.

Paragraph tags are or are not allowed depending on many conditions. Section 51 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a \LaTeX paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output: 4671 `\begin{warpHTML}`

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

```

4672 \newcounter{LWR@spandepth}
4673 \setcounter{LWR@spandepth}{0}

```

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

```

4674 \newbool{LWR@doingstartpars}
4675 \boolfalse{LWR@doingstartpars}

```

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

```

4676 \newbool{LWR@doingapar}
4677 \global\boolfalse{LWR@doingapar}

```

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

```

4678 \newcommand*{\LWR@ensuredoingapar}{%
4679 \ifbool{LWR@doingstartpars}%
4680 {\global\booltrue{LWR@doingapar}}%
4681 {}}%
4682 }

```

`\PN@parnotes@auto` Redefined by **parnotes** to print paragraph notes at the end of each paragraph.

```

4683 \def\Pn@parnotes@auto{}%

```

`\LWR@openparagraph`

```

4684 \newcommand*{\LWR@openparagraph}
4685 {%

```

See if paragraph handling is enabled:

```

4686 \ifbool{LWR@doingstartpars}%
4687 {% handling pars

```

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

```

4688 \ifbool{expr{
4689     test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
4690     test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
4691 }% 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 ``.

```

4692 }% no nested par tags

```

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

```

4693 {% yes nest par tags

```

If **parnotes** is used, paragraph notes are inserted before starting the next paragraph:

```
4694      \PN@parnotes@auto%
```

The opening paragraph tag:

```
4695      \LWR@htmltagc{\LWR@tagregularparagraph}%
```

Now have started a paragraph.

```
4696      \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.)

```
4697      \let\par\LWR@closeparagraph%
4698      }% end of yes nest par tags
4699 }% end of handling pars
4700 {}% not handling pars
4701 }
```

`\LWR@closeparagraph`

```
4702 \newcommand*{\LWR@closeparagraph}
4703 {%
4704 % \LWR@traceinfo{LWR@closeparagraph}%
```

See if paragraph handling is enabled:

```
4705 \ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
4706 {% handling pars
```

See if already started a `lateximage` or a ``:

```
4707 \ifboolexpr{
4708     test {\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}} or
4709     test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}}
4710 }%
```

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

```
4711     {% no nested par tags
```

```

4712     \ifboolexpr{
4713         test {\ifnumcomp{\value{LWR@spandepth}}{>}{0}} and
4714         test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
4715     }%
4716     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
4717     {}%
4718 }% no nested par tags

```

If have not already started a lateximage or a :

```

4719     {% yes nest par tags

```

Print a closing tag and some extra vertical space.

```

4720     \unskip%
4721     \LWR@htmltagc{/LWR@tagregularparagraph}%
4722     \LWR@orignewline%

```

No longer doing a paragraph:

```

4723     \global\boolfalse{LWR@doingapar}%

```

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

```

4724     \global\boolfalse{LWR@minipagethispar}%

```

If **parnotes** is used, paragraph notes are inserted after ending the previous paragraph:

```

4725     \PN@parnotes@auto%
4726     }% end of yes nest par tags
4727 }% end of handling pars

```

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

```

4728 {% not handling pars
4729     \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
4730     {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
4731     {}%
4732 }% 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:

```

4733 \ifboolexpr{%
4734     not bool {LWR@doingapar} and
4735     test {\ifnumcomp{\value{LWR@tabulardepth}}{>}{0}} and
4736     test {

```

```

4737         \ifnumcomp{\value{LWR@tabulardepth}}{=}{\value{LWR@tabularpardepth}}
4738     } and
4739     bool {LWR@intabularmetadata} and
4740     not bool {LWR@tableparcell} and
4741     test {\ifnumcomp{\value{LWR@lateximagedepth}}{=}{0}}
4742 }%
4743 {%
4744     \LWR@getmynexttoken%
4745 }{%
4746     \LWR@origpar%
4747 }%
4748 }

4749 \end{warpHTML}

```

51 Paragraph start/stop handling

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

Section 50 has the commands which actually generate the tags.

The everyhook package is used to generate the opening paragraph tags. The closing tags are generated by `\par`.

for HTML output: 4750 `\begin{warpHTML}`

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

```

4751 \newcommand*{\LWR@startpars}%
4752 {%
4753 % \LWR@traceinfo{\LWR@startpars}%

```

Ignore if inside a span:

```

4754 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%
4755 }%
4756 {%

```

See if currently handling HTML paragraphs:

```

4757     \ifbool{LWR@doingstartpars}%

```

If already in paragraph mode, do nothing.

4758 { }%

If not currently in paragraph mode:

4759 { %

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

4760 \PushPreHook{par}{\LWR@openparagraph}%

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

4761 \let\par\LWR@closeparagraph

4762

4763 }% an intentionally blank line

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

4764 \global\setbool{LWR@doingstartpars}{true}%

No <par> tag yet to undo:

4765 \global\boolfalse{LWR@doingapar}%

4766 }% nestspan

4767 % \LWR@traceinfo{LWR@startpars: done}%

4768 }

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

4769 \newcommand*{\LWR@stoppars}%

4770 {%

Ignore if inside a span:

4771 \ifnumcomp{\value{LWR@spandepth}}{>}{0}%

4772 {}%

4773 {%

See if currently handling HTML paragraphs:

4774 \ifbool{LWR@doingapar}%

if currently in an HTML paragraph:

4775 {%

Print a closing tag:

```
4776      \unskip%
4777      \LWR@htmltagc{/\LWR@tagregularparagraph}%
4778      \LWR@orignewline%
```

No longer have an open HTML paragraph:

```
4779      \global\boolfalse{LWR@doingapar}%
```

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

```
4780      \global\boolfalse{LWR@minipagethispar}
4781
4782      }% an intentionally blank line
```

If was not in an HTML paragraph:

```
4783      {}%
```

See if currently allowing HTML paragraphs:

```
4784      \ifbool{LWR@doingstartpars}%
```

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

```
4785      {\ClearPreHook{par}}%
```

Else: Do nothing:

```
4786      {}%
```

No longer in paragraph mode:

```
4787      \global\setbool{LWR@doingstartpars}{false}%
```

No <p> tag to undo:

```
4788      \global\boolfalse{LWR@doingapar}%
4789      }% nestspan
4790      }
```

```
4791 \end{warpHTML}
```

52 Page headers and footers

for HTML & PRINT: 4792 \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.

```
4793 \newcommand{\LWR@firstpagetop}{} % for the home page alone
4794 \newcommand{\LWR@pagetop}{} % for all other pages
4795 \newcommand{\LWR@pagebottom}{}

```

\HTMLFirstPageTop {<*text and logos*>}

```
4796 \newcommand{\HTMLFirstPageTop}[1]{%
4797     \renewcommand{\LWR@firstpagetop}{#1}%
4798 }

```

\HTMLPageTop {<*text and logos*>}

```
4799 \newcommand{\HTMLPageTop}[1]{%
4800     \renewcommand{\LWR@pagetop}{#1}%
4801 }

```

\HTMLPageBottom {<*text and logos*>}

```
4802 \newcommand{\HTMLPageBottom}[1]{%
4803     \renewcommand{\LWR@pagebottom}{#1}%
4804 }

```

```
4805 \end{warpall}

```

53 CSS

for HTML output: 4806 \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.

```
4807 \newcommand*{\LWR@currentcss}{lwarp.css}

```

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

```
4808 \newcommand*{\CSSFilename}[1]{%
4809 \renewcommand*{\LWR@currentcss}{#1}%
4810 \@onelevel@sanitize\LWR@currentcss%
4811 }
4812
4813 \end{warpHTML}
```

for PRINT output: 4814 `\begin{warpprint}`
 4815 `\newcommand*{\CSSFilename}[1]{}`
 4816 `\end{warpprint}`

54 Title, HTML meta author, HTML meta description

for HTML output: 4817 `\begin{warpHTML}`

`\title` `{\title}` Modified to remember `\thetitle`, which is used to set the HTML page titles.

```
4818 \let\LWR@origtitle\title
4819
4820 \renewcommand*{\title}[1]{%
4821   \LWR@origtitle{#1}%
4822   \begingroup%
4823     \renewcommand{\thanks}[1]{}%
4824     \protected@xdef\thetitle{#1}%
4825   \endgroup%
4826 }
4827 \end{warpHTML}
```

for HTML & PRINT: 4828 `\begin{warppall}`

`\HTMLTitle` `{\Titlename}` The Title to place into an HTML meta tag. The default is to use the document `\title`'s setting.

```
4829 \providecommand{\thetitle}{}
4830
4831 \newcommand{\theHTMLTitle}{\thetitle}
4832
4833 \newcommand{\HTMLTitle}[1]{\renewcommand{\theHTMLTitle}{#1}}
```

`\HTMLAuthor` `{\authorname}` The author to place into an HTML meta tag. If none given, the default is `\theauthor`, which is empty unless the **titling** package is used.

```
4834 \providecommand{\theauthor}{}
4835
4836 \newcommand{\theHTMLAuthor}{\theauthor}
4837
4838 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}
```

This is placed inside an HTML meta tag at the start of each file. This may be changed mid-document using `\HTMLDescription`, allowing different HTML descriptions to be used for different sections of the document.



Do not use double quotes, and do not exceed 150 characters.

`\HTMLDescription` `{\New HTML meta description.}` Assigns the HTML file's description meta tag.

```
4839 \newcommand{\LWR@currentHTMLDescription}{}
4840
4841 \newcommand{\HTMLDescription}[1]{%
4842 \renewcommand{\LWR@currentHTMLDescription}{#1}
4843 }
4844
4845 \end{warpall}
```

55 Footnotes

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

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



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

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

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

```

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

```

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

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

55.1 Regular page footnotes

In HTML documents, footnotes are placed at the bottom of the web page or the section, depending on `FootnoteDepth`, using the \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 55.4 for the implementation.

55.2 Minipage footnotes

See section 55.5 for how minipage footnotes are gathered. See section 85.3 for how minipage footnotes are placed into the document.

55.3 Titlepage thanks

See section 62.7 for titlepage footnotes.

55.4 Regular page footnote implementation

for HTML & PRINT: 4846 `\begin{warpall}`

Ctr	<code>FootnoteDepth</code>	Determines how deeply to place footnotes in the HTML files, similar to <code>tocdepth</code> . Default: 3
		The default of 3 places footnotes before each <code>\subsubsection</code> or higher. See table 7 for a table of \TeX section headings.

```

4847 \newcounter{FootnoteDepth}
4848 \setcounter{FootnoteDepth}{3}

4849 \end{warpall}

```

for HTML output: 4850 `\begin{warpHTML}`

Patch \LaTeX footnotes to use a new `\box` instead of an insert for **lwarp** footnotes. This avoids having the original `\footins` appear at the bottom of a `lateximage`, which is on its own new page.

4851 `\newbox\LWR@footnotes`

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

`\@makefntext` `{\text}`

4852 `\long\def\@makefntext#1{\@thefnmark~#1}`

`\@makefnmark`

```
4853 \def\@makefnmark{%
4854     \textsuperscript{\@thefnmark}%
4855 }
```

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}`

```
4856 \long\def\LWR@footnotetext#1{%
4857 \LWR@traceinfo{\LWR@footnotetext}%
4858 \global\setbox\LWR@footnotes=\vbox{%
```

Add to any current footnotes:

4859 `\unvbox\LWR@footnotes%`

Remember the footnote number for `\ref`:

```
4860 \protected@edef\@currentlabel{%
4861     \csname p@footnote\endcsname\@thefnmark%
4862 }% \@currentlabel
```

Open a group:

```
4863 \color@begingroup%
```

Use HTML superscripts in the footnote even inside a lateximage:

```
4864 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

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

```
4865 \ifthenelse{%
4866     \boolean{LWR@doingstartpars} \AND%
4867     \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
4868 }%
4869 {}%
4870 {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
4871 \@makefntext{#1}%
```

Closing paragraph tag:

```
4872 \ifthenelse{%
4873     \boolean{LWR@doingstartpars} \AND%
4874     \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
4875 }%
4876 {\par}%
4877 {%
4878     \LWR@htmltagc{/LWR@tagregularparagraph}%
4879     \LWR@orignewline%
4880 }%
```

Close the group:

```
4881 \color@endgroup%
4882 }% vbox
```

Paragraph handling:

```
4883 \LWR@ensuredoingapar%
4884 }%
```

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

```
4885 \LetLtxMacro\@footnotetext\LWR@footnotetext
```

55.5 Minipage footnote implementation

Patch \LaTeX minipage footnotes to use a new `\box` instead of an insert for **lwarp** minipage footnotes. This avoids having the original `\@mpfootins` appear at the bottom of a `lateximage`, which is on its own new page.

```
4886 \newbox\LWR@mpfootnotes
```

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

```
4887 \long\def\@mpfootnotetext#1{%
4888 \LWR@traceinfo{\@mpfootnotetext}%
4889 \global\setbox\LWR@mpfootnotes\vbox{%
4890   \unvbox\LWR@mpfootnotes%
4891   \reset@font\footnotesize%
4892   \hsize\columnwidth%
4893   \@parboxrestore%
4894   \protected@edef\@currentlabel%
4895     {\csname p@mpfootnote\endcsname\@thefnmark}%
4896   \color@begingroup%
```

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

```
4897   \ifthenelse{%
4898     \boolean{LWR@doingstartpars} \AND%
4899     \cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
4900   }%
4901   {}%
4902   {\LWR@htmltagc{\LWR@tagregularparagraph}}%

4903   \@makefntext{%
4904     \ignorespaces#1%
4905   }%
```

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

```
4906   \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
4907   {}%
4908   {%
4909     \LWR@htmltagc{/\LWR@tagregularparagraph}%
4910     \LWR@orignewline%
4911   }%
4912   \color@endgroup%
4913 }% vbox
```

Paragraph handling:

```

4914 \LWR@ensuredoingapar%
4915 \LWR@traceinfo{@mpfootnotetext: done}%
4916 }

```

`\thempfootnote` Redefined to remove the `\itshape`, which caused an obscure compiling error in some situations.

```

4917 \AtBeginDocument{
4918 \def\thempfootnote{\@alph\c@mpfootnote}
4919 }

```

55.6 Printing pending footnotes

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

```

4920 \newcommand*{\LWR@printpendingfootnotes}{%
4921 \ifvoid\LWR@footnotes\else
4922   \LWR@forcenewpage
4923   \begin{BlockClass}{footnotes}
4924   \LWR@origmedskip
4925   \unvbox\LWR@footnotes
4926   \setbox\LWR@footnotes=\vbox{}
4927   \end{BlockClass}
4928 \fi
4929 }

```

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

```

4930 \newcommand*{\LWR@maybeprintpendingfootnotes}[1]{%
4931 \ifboolexpr{
4932   not test{\ifnumcomp{#1}{>}{\value{FootnoteDepth}}} or
4933   bool{FormatEPUB} or
4934   bool{FormatWP}
4935 }%
4936 {\LWR@printpendingfootnotes}%
4937 {}%
4938 }

```

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

```

4939 \newcommand*{\LWR@printpendingmpfootnotes}{%
4940 \ifvoid\LWR@mpfootnotes\else
4941   \LWR@forcenewpage

```

```

4942 \begin{BlockClass}{footnotes}
4943 \LWR@print@vspace*{\baselineskip}
4944 \unvbox\LWR@mpfootnotes
4945 \setbox\LWR@mpfootnotes=\vbox{}
4946 \end{BlockClass}
4947 \fi
4948 }

4949 \end{warpHTML}

```

56 Marginpars

`\marginpar` [*<left>*] [*<right>*] `\marginpar` may contains paragraphs, but in order to remain inline with the surrounding text **lwarp** nullifies block-related macros inside the `\marginpar`. Paragraph breaks are converted to `
` 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: 4950 `\begin{warpHTML}`

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

```

4951 \renewcommand{\marginpar}[2][]{%
4952 \ifbool{FormatWP}%
4953 {%
4954 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{\marginblock}
4955 #2
4956 \end{LWR@BlockClassWP}
4957 }%
4958 {%
4959 \LWR@htmlspanclass{marginpar}{#2}%
4960 }%
4961 }

```

`\marginparBlock` [*<left>*] [*<right>*]

For use when the marginpar will be more than one paragraph, and/or contains more than simple text.

HTML version.

```

4962 \newcommand{\marginparBlock}[2][]{%

```

```

4963 \ifbool{FormatWP}%
4964 {%
4965 \begin{LWR@BlockClassWP}{width:2in; float:right; margin:10pt}{}{marginblock}
4966 #2
4967 \end{LWR@BlockClassWP}
4968 }%
4969 {%
4970 \begin{BlockClass}[width:2in; float:right; margin:10pt]{marginparblock}
4971 #2
4972 \end{BlockClass}
4973 }%
4974 }

```

`\reversemarginpar`

```
4975 \renewcommand*{\reversemarginpar}{}

```

`\normalmarginpar`

```
4976 \renewcommand*{\normalmarginpar}{}

```

```
4977 \end{warpHTML}

```

for PRINT output: 4978 `\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.

```
4979 \LetLtxMacro\marginparBlock\marginpar

```

```
4980 \end{warpprint}

```

57 Splitting HTML files

- Files are split according to FileDepth and CombineHigherDepths.
- Filenames are sanitized by `\LWR@filenamenoblanks`.
- `\LWR@newhtmlfile` finishes an HTML page, adds a comment to tell where and how to split the file, then starts a new HTML page.

for HTML & PRINT: 4981 \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.

```
4982 \newcounter{FileDepth}
4983 \setcounter{FileDepth}{-5}
```

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

```
4984 \newbool{CombineHigherDepths}
4985 \booltrue{CombineHigherDepths}
```

```
4986 \end{warpall}
```

for HTML output: 4987 \begin{warpHTML}

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

```
4988 \newcommand*{\LWR@thisfilename}{}
```

\LWR@thisnewfilename The filename being sanitized.

```
4989 \newcommand*{\LWR@thisnewfilename}{}
```

\LWR@filenamenoblanks {\<filename>}

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



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

```
4990 \newcommand*{\LWR@filenamenoblanks}[1]{%
4991 \begingroup
```

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

```
4992 \LWR@nullfonts%
4993 \renewcommand*{\LWR@htmltagc}[1]{}%
```

Replaces common symbols and short words with hyphens:

```
4994 \edef\LWR@thisnewfilename{#1}%
4995 \LWR@traceinfo{\LWR@filenamenoblanks} \edef: !\LWR@thisnewfilename!}%
4996 \fullexpandarg%
```

Convert spaces into hyphens:

```
4997 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
4998 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
4999 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
5000 \StrSubstitute{\LWR@thisnewfilename}%
5001     {\LWR@origampersand}{-}[\LWR@thisnewfilename]
5002 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
5003 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
5004 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
5005 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
5006 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
5007 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
5008 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
5009 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
5010 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
5011 \StrSubstitute{\LWR@thisnewfilename}%
5012     {\textless}{-}[\LWR@thisnewfilename]
5013 \StrSubstitute{\LWR@thisnewfilename}%
5014     {\textgreater}{-}[\LWR@thisnewfilename]
5015 \StrSubstitute{\LWR@thisnewfilename}{\LWR@origpound}{-}[\LWR@thisnewfilename]

5016 \StrSubstitute{\LWR@thisnewfilename}{\_}{-}[\LWR@thisnewfilename]

5017 \StrSubstitute{\LWR@thisnewfilename}{\ }{-}[\LWR@thisnewfilename]
5018 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
5019 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
5020 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
5021 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
5022 \StrSubstitute{\LWR@thisnewfilename}%
5023     {\textbackslash}{-}[\LWR@thisnewfilename]
5024 \StrSubstitute{\LWR@thisnewfilename}{^}{-}[\LWR@thisnewfilename]
5025 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
5026 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
5027 %     "~{}" for babel
5028 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
5029 \StrSubstitute{\LWR@thisnewfilename}{[]}{-}[\LWR@thisnewfilename]
5030 \StrSubstitute{\LWR@thisnewfilename}{' }{-}[\LWR@thisnewfilename]
```

Convert short words:

```
5031 \StrSubstitute{\LWR@thisnewfilename}{-s}{-}[\LWR@thisnewfilename]
5032 \StrSubstitute{\LWR@thisnewfilename}{-S}{-}[\LWR@thisnewfilename]
5033 \StrSubstitute{\LWR@thisnewfilename}{-a}{-}[\LWR@thisnewfilename]
5034 \StrSubstitute{\LWR@thisnewfilename}{-A}{-}[\LWR@thisnewfilename]
```

```

5035 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
5036 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
5037 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
5038 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
5039 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
5040 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
5041 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
5042 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
5043 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
5044 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
5045 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
5046 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
5047 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
5048 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

5049 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
5050 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
5051 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
5052 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]

```

If pdf_{La}TeX and not utf8 encoding, don't try to convert emdash, endash:

```

5053 \ifPDFTeX%
5054 \ifdefstring{\inputencodingname}{utf8}{%
5055 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
5056 %      emdash
5057 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
5058 %      endash
5059 }{}%
5060 \else% not PDFTeX
5061 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
5062 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
5063 \fi%

```

Return the result:

```

5064 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
5065 \endgroup%
5066 \LWR@traceinfo{\LWR@filenamenoblanks: result is \LWR@thisfilename}%
5067 }

```

Ctrl Remembers which autopage label was most recently generated. Used to avoid duplicates.

```

5068 \newcounter{\LWR@previousautopagelabel}
5069 \setcounter{\LWR@previousautopagelabel}{-1}

```

`\LWR@newautopagelabel` $\{\langle\textit{pagenumber counter}\rangle\}$

```

5070 \newcommand*{\LWR@newautopagelabel}[1]{%
5071 \ifnumequal{\value{\LWR@previousautopagelabel}}{\value{page}}%
5072 {}% no action if this autopage label has already been defined
5073 {%
5074   \label{autopage-\arabic{#1}}%
5075   \setcounter{\LWR@previousautopagelabel}{\value{page}}
5076 }%
5077 }
```

`\LWR@customizedMathJax` Additional MATHJAX definitions to be added to the start of each HTML page.

```

5078 \newcommand*{\LWR@customizedMathJax}{}
```

`\CustomizeMathJax` MATHJAX does not have preexisting support every possible math function. Additional MATHJAX function definitions may be defined. These will be declared at the start of each HTML page, and thus will have a global effect.

Examples:

```

\CustomizeMathJax{
  \newcommand{\expval}[1]{\langle#1\rangle}
  \newcommand{\abs}[1]{\lvert#1\rvert}
}
\CustomizeMathJax{\newcommand{\arsinh}{\text{arsinh}}}
\CustomizeMathJax{\newcommand{\arcosh}{\text{arcosh}}}
\CustomizeMathJax{\newcommand{\NN}{\mathbb{N}}}

5079 \newcommand*{\CustomizeMathJax}[1]{%
5080   \appto{\LWR@customizedMathJax}{%
5081     \(#1\)\par
5082   }%
5083 }
```

`\LWR@customizeMathJax`

```

5084 \newcommand{\LWR@customizeMathJax}{%
5085 \ifbool{mathjax}{
5086 \LWR@stoppars
5087 \LWR@htmlcomment{Nullify \textbackslash{}ensuremath for MathJax:}
5088
5089 \(\newcommand\ensuremath[1]{\##1}\)
5090
5091 \(\newcommand\footnote[2][]{\text{( Footnote ##1 )}}\()
5092
5093 \(\newcommand\footnotemark[1][]{\text{( Footnote ##1 )}}\()
```

```

5094
5095 \LWR@htmlcomment{Additional customizations for MathJax:}
5096
5097 \LWR@customizedMathJax
5098
5099 \LWR@startpars
5100 }{}
5101 }

```

```

5102 \end{warpHTML}

```

for PRINT output: 5103 \begin{warpprint}

\CustomizeMathJax The print-mode version:

```

5104 \newcommand*{\CustomizeMathJax}[1]{}
5105 \end{warpprint}

```

for HTML output: 5106 \begin{warpHTML}

\LWR@newhtmlfile {<section name>}

Finishes the current HTML page with footnotes, footer, navigation, then starts a new HTML page with an HTML comment telling where to split the page and what the new filename and CSS are, then adds navigation, side TOC, header, and starts the text body.

```

5107 \newcommand*{\LWR@newhtmlfile}[1]{
5108 \LWR@traceinfo{\LWR@newhtmlfile}

```

At the bottom of the ending file:

```

5109 \LWR@htmlclassend{section}{textbody}
5110
5111 \LWR@printpendingfootnotes
5112

```

No footer between files if EPUB:

```

5113 \ifbool{FormatEPUB}
5114 {}
5115 {
5116     \LWR@htmlclassend{footer}
5117
5118     \LWR@pagebottom

```

```

5119
5120     \LWR@html elementend{footer}
5121 }

```

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

```

5122 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5123 {}
5124 {\ifnumcomp{\value{LWR@htmlfile number}}{>}{0}{\LWR@botnavigation}{}}

```

End of this HTML file:

```

5125 \LWR@stoppars
5126 \LWR@htmltag{/body}\LWR@orignewline
5127 \LWR@htmltag{/html}\LWR@orignewline
5128 \LWR@traceinfo{LWR@newhtmlfile: about to LWR@orignewpage}
5129 \LWR@orignewpage
5130
5131 \addtocounter{LWR@htmlfile number}{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.

```

5132 \ifbool{FileSectionNames}%
5133 {\LWR@filenameno blanks{#1}}
5134 {\renewcommand*{\LWR@thisfilename}{\arabic{LWR@htmlfile number}}}

```

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.

```

5135 \LWR@traceinfo{LWR@newhtmlfile: about to print start file}%

```

\LWR@nullfonts to allow math in a section name.

```

5136 \begingroup%
5137 \LWR@nullfonts%
5138 \LWR@htmlblockcomment{%
5139 |Start file|
5140 \LWR@htmlsectionfilename{\LWR@thisfilename}|
5141 }
5142 \endgroup%

```

At the top of the starting file:

```
5143 \LWR@stoppars
5144
```

If pdf_{La}TeX and not utf8 encoding, use a hyphen instead of an emdash:

```
5145 \ifPDFTeX%
5146 \ifdefstring{\inputencodingname}{utf8}{%
5147 \LWR@filestart{ -- #1}% there is an EMDash in front of the #1
5148 }{
5149 \LWR@filestart{ - #1}% hyphen
5150 }
5151 \else%
5152 \LWR@filestart{ -- #1}% there is an EMDash in front of the #1
5153 \fi%
5154
```

Track the page numbers:

```
5155 \setcounter{LWR@latestautopage}{\value{page}}%
5156 \LWR@newautopagelabel{LWR@latestautopage}%
```

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

```
5157 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5158 {}
5159 {\LWR@topnavigation}
5160
```

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

```
5161 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5162 {}
5163 {
5164     \LWR@htmlelement{header}
5165
5166     \LWR@pagetop
5167
5168     \LWR@htmlelementend{header}
5169 }
5170
```

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

```
5171 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5172 {}
5173 {\ifcvoid{thetitle}{ }\LWR@printthetitle}
5174
```

No sidetoc if formatting for an EPUB or word processor:

```
5175 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5176 {}
5177 {\LWR@sidetoc}
5178
```

Start of the <tbody>:

```
5179 \LWR@htmlclass{section}{tbody}
5180
```

Keep paragraph tags disabled for now:

```
5181 \LWR@stoppars
5182
```

If using MathJax, disable `\ensuremath` by printing a nullified definition at the start of each file, and add further customizations:

```
5183 \LWR@customizeMathJax

5184 \LWR@traceinfo{LWR@newhtmlfile: done}
5185 }

5186 \end{warpHTML}
```

58 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing \TeX code and packages. Formatting is handled by CSS, so the emulated code has much less work to do than the print versions.

Unicode Section names and the resulting filenames with accented characters are partially supported, depending on the ability of **pdf \LaTeX** to generate characters and **pdf tex** to read them. If extra symbols appear in the text, it may be that **pdf \LaTeX** is actually producing a symbol over or under a character, resulting in **pdf tex** picking up the accent symbol separately.



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

for HTML output: 5187 \begin{warpHTML}

58.1 User-level starred section commands

`\ForceHTMLPage` For HTML output, forces the next section to be on its own HTML page, if `FileDepth` allows, even if starred. For use with `\printindex` and others which generate a starred section which should be on its own HTML page. Also see `\ForceHTMLTOC`.

For print output, no effect.

```
5188 \newbool{LWR@forcinghtmlpage}
5189 \boolfalse{LWR@forcinghtmlpage}
5190
5191 \newcommand*{\ForceHTMLPage}{%
5192 \global\booltrue{LWR@forcinghtmlpage}%
5193 }
```

`\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.

```
5194 \newbool{LWR@forcinghtmltoc}
5195 \boolfalse{LWR@forcinghtmltoc}
5196
5197 \newcommand*{\ForceHTMLTOC}{%
5198 \global\booltrue{LWR@forcinghtmltoc}%
5199 }
```

```
5200 \end{warpHTML}
```


for PRINT output:

```
5201 \begin{warpprint}
5202 \newcommand*{\ForceHTMLPage}{}
5203 \newcommand*{\ForceHTMLTOC}{}
5204 \end{warpprint}
```

for HTML output:

```
5205 \begin{warpHTML}
```

58.2 Book class commands

`\mainmatter`  Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.

```
5206 \newbool{LWR@mainmatter}
```

```
5207 \DeclareDocumentCommand{\mainmatter}{-}{-}{%
5208 \booltrue{LWR@mainmatter}%
5209 }
```

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

```
5210 \DeclareDocumentCommand{\frontmatter}{-}{-}{%
5211 \boolfalse{LWR@mainmatter}%
5212 }
```

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

```
5213 \DeclareDocumentCommand{\backmatter}{-}{-}{%
5214 \boolfalse{LWR@mainmatter}
5215 }
```

58.3 Sectioning support macros

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

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

```
5216 \newcommand*{\LWR@sectionnumber}[1]{%
5217 \InlineClass{sectionnumber}{#1}%
5218 }
```

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

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

Create an autosection tag.

```
5219 \newcommand*{\LWR@createautosec}[1]{%
5220 \LWR@htmltag{#1 id="\LWR@print@mbbox{autosec-\arabic{page}}"}%
5221 }
```

`\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.

```
5222 \NewDocumentCommand{\LWR@pushoneclose}{m m}{%

```

```

5223 \LWR@traceinfo{\LWR@pushoneclose #1}%
5224     \pushclose{#2}{#1}%
5225 }

```

`\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.

```

5226 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%

```

Close any stacked sections up to this new one.

```

5227 \LWR@closeprevious{#1}%

```

Push a new section depth:

```

5228 \LWR@pushoneclose{#1}{#2}%
5229 }

```

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.

```

5230 \newcounter{LWR@prevFileDepth}
5231 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}

```

`\@secCNTformat` $\{\langle sectiontype \rangle\}$

```

5232 \def\@secCNTformat#1{\csname the#1\endcsname\protect\quad}

```

`\simplechapterdelim` Used by **tocbibind** and **anonchap**.

```

5233 \newcommand*{\simplechapterdelim}{\}

```

`\@chapCNTformat` $\{\langle sectiontype \rangle\}$

`\let` to `\@secCNTformat` by default, but may be redefined by `\simplechapter` and `\restorechapter` from **tocbibind** or **anonchap**.

```

5234 \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.

```
5235 \newcounter{LWR@currentautosec}
```

```
\LWR@section * [{<TOC name>}] {<name>} {<sectiontype>}
```

The common actions for the high-level sectioning commands.

```
5236 \DeclareDocumentCommand{\LWR@section}{m m m m}{%
5237 \LWR@traceinfo{LWR@section |#2| |#3|}%
5238 \LWR@traceinfo{LWR@section: not an empty section}%
5239 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
5240 \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:

```
5241 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
5242 \IfBooleanT{#1}{\LWR@traceinfo{LWR@section: starred}}}%
5243 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{}}}%
5244 \ifthenelse{%
5245   \(%
5246     \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
5247     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
5248     \(\boolean{LWR@forcinghtmlpage}\)\)%
5249   \)%
5250   \AND%
5251   \cnttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}}%
5252   \AND%
5253   \(%
5254     \NOT\boolean{CombineHigherDepths}\OR%
5255     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}}%
5256   \)%
5257   \AND%

5258   \(% phantomsection
5259     \NOT\isempty{#3}%
5260     \OR%
5261     \(\NOT\equal{#1}{\BooleanTrue}\)\)%
```

```
5262    \)%
5263 }%
```

If so: start a new HTML file:

```
5264 {% new file
5265    \LWR@traceinfo{LWR@section: new HTML file}%
```

See if there was an optional TOC name entry:

```
5266    \IfNoValueTF{#2}%
```

If no optional entry

```
5267    {\LWR@newhtmlfile{#3}}%
```

If yes an optional entry

```
5268    {\LWR@newhtmlfile{#2}}%
5269 }% new file
```

Else: No new HTML file:

```
5270 {% not new file
```

Generate a new \TeX page so that toc and index page number points to the section:

```
5271    \LWR@traceinfo{LWR@section: not a new HTML file, about to LWR@orignewpage}%
5272    \LWR@orignewpage%
5273
5274 }% not new file
```

Remember this section's name for \nameref:

```
5275 \IfValueT{#3}{%
5276 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
5277 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
5278 }%
```

Print an opening comment with the level and the name; ex: “section” “Introduction”
Footnotes may be used in section names, which would also appear in the HTML
section opening comments, so the short toc entry is used if possible, and a limited
opening comment is made if the sectional unit is starred.

```
5279
5280 \ifbool{HTMLDebugComments}{%
5281    \begingroup%
```

```

5282 \LWR@nullfonts%
5283 \IfBooleanTF{#1}% starred
5284 {\LWR@htmlcomment{Opening #4*}}%
5285 {%
5286 \IfNoValueTF{#2}% short TOC
5287 {\LWR@htmlcomment{Opening #4 ‘‘#3’’}}%
5288 {\LWR@htmlcomment{Opening #4 ‘‘#2’’}}%
5289 }
5290 \endgroup%
5291 }{}%
5292

```

For inline sections paragraph and subparagraph, start a new paragraph now:

```

5293 \ifthenelse{%
5294 \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
5295 }%
5296 {\LWR@startpars}%
5297 {}%

```

Create the opening tag with an autosec:

```

5298 \LWR@traceinfo{LWR@section: about to LWR@createautosec}%
5299 \LWR@createautosec{\csuse{LWR>tag#4}}%

5300 \setcounter{LWR@currentautosec}{\value{page}}

```

Check if starred:

```

5301 \IfBooleanTF{#1}%
5302 {%
5303 \LWR@traceinfo{LWR@section: starred}%

```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```

5304 \ifbool{LWR@forcinghtmltoc}%
5305 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
5306 {}%
5307 }% starred

```

Not starred, so step counter and add to TOC:

```

5308 {% not starred

```

Only add a numbered TOC entry if section number is not too deep:

```

5309     \ifthenelse{%
5310         \cntttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
5311     }%
5312     {% if secnumdepth

```

If in the main matter, step the counter and add the TOC entry. For article class, **lwarp** assumes that all is mainmatter.

```

5313         \LWR@traceinfo{LWR@section: about to test main matter}%
5314         \ifbool{LWR@mainmatter}%
5315     {%
5316         \LWR@traceinfo{LWR@section: yes mainmatter}%
5317         \refstepcounter{#4}%

```

Add main matter numbered TOC entry with the TOC name or the regular name:

```

5318         \LWR@traceinfo{LWR@section: about to addcontentsline}%
5319         \addcontentsline{toc}{#4}%
5320     {%
5321         \protect\numberline{\csuse{the#4}}%
5322         {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
5323     }%
5324         \LWR@traceinfo{LWR@section: finished addcontentsline}%
5325     }% end of if main matter

```

If not main matter, add unnumbered TOC name or regular name:

```

5326     {% not main matter
5327         \LWR@traceinfo{LWR@section: no main matter}%
5328         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
5329     }% end of not main matter
5330 }% end of secnumdepth

```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```

5331     {%
5332         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
5333     }%

```

For part, print the section type:

```

5334     \ifbool{LWR@mainmatter}%
5335     {%
5336         \ifthenelse{%
5337             \(\cntttest{\csuse{LWR@depth#4}}{<=}%
5338             {\value{secnumdepth}}\)\ AND%
5339             \(\cntttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}}\)%
5340     }%

```

```
5341      {\csuse{#4name}~{}}%
5342      {}}%
```

Print the section number:

```
5343      \LWR@traceinfo{LWR@section: about to print section number}%
5344      \ifthenelse{%
5345          \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
5346      }%
5347      {%
5348          \ifstrequal{#4}{chapter}%
5349          {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%
5350          {\protect\LWR@sectionnumber{\@secCNTformat{#4}}}%
5351      }%
5352      {}}%
5353      \LWR@traceinfo{LWR@section: finished print section number}%
5354      {}}%
5355 }% end of not starred
```

Print the section name:

```
5356 \LWR@traceinfo{LWR@section: about to print the section name}%
5357 #3%
```

Close the heading tag, such as /H2:

```
5358 \LWR@traceinfo{LWR@section: about to close the heading tag}%
5359 \LWR@htmltag{\csuse{LWR@tag#4end}}%
```

Generate a \LaTeX label:

```
5360 \LWR@traceinfo{LWR@section: about to create the LaTeX label}%
5361 \LWR@newautopagelabel{LWR@currentautosec}%
```

Start paragraph handing unless is an inline paragraph or subparagraph:

```
5362 \ifthenelse{%
5363     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
5364 }%
5365 {\LWR@startpars}%
5366 {}}%
```

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.

```

5367 \ifthenelse{%
5368     \NOT\equal{#1}{\BooleanTrue}\OR%
5369     \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
5370 }%
5371 {% not starred
5372     \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
5373 }% not starred
5374 {}%
```

Reset to defaults if not a phantomsection:

```

5375 \ifstrempy{#3}%
5376 {}%
5377 {%
5378 \global\boolfalse{LWR@forcinghtmlpage}%
5379 \global\boolfalse{LWR@forcinghtmltoc}%
5380 }%
5381 %
5382 \LWR@traceinfo{LWR@section: done}%
5383 }
```

58.4 \section and friends

`\part` * [*TOC name*] {*<name>*}

```

5384 \newcommand{\part@preamble}{}% for koma-script
5385
5386 \DeclareDocumentCommand{\part}{s o m}{%
5387 \LWR@maybeprintpendingfootnotes{\LWR@depthpart}%
5388 \LWR@stoppars%
5389
5390 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
5391
5392 \LWR@section{#1}{#2}{#3}{part}%
5393
5394 \part@preamble% for koma-script
5395 \renewcommand{\part@preamble}{}%
5396 }
```

`\chapter` * [*TOC name*] [*<heading name>*] {*<name>*}

```

5397 \let\@printcites\relax% for quotchap package
5398
5399 \newcommand{\chapter@preamble}{}% for koma-script
5400
5401 \@ifundefined{chapter}
5402 {}
5403 {%
5404 \DeclareDocumentCommand{\chapter}{s o m}{%
5405 \IfValueTF{#2}{
5406 \LWR@traceinfo{chapter #2}%
5407 }{
5408 \LWR@traceinfo{chapter #4}%
5409 }
5410 \LWR@maybeprintpendingfootnotes{\LWR@depthchapter}%
5411 \LWR@stoppars%
5412
5413 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
5414
5415 \LWR@section{#1}{#2}{#4}{chapter}%
5416
5417 \@printcites% for quotchap package
5418
5419 \chapter@preamble% for koma-script
5420 \renewcommand{\chapter@preamble}{}%
5421 }
5422 }

```

`\section * [⟨TOC name⟩] [⟨heading name⟩] {⟨name⟩}`

```

5423 \DeclareDocumentCommand{\section}{s o m}{%
5424 \IfValueTF{#2}{
5425 \LWR@traceinfo{section #2}%
5426 }{
5427 \LWR@traceinfo{section #4}%
5428 }
5429 \LWR@maybeprintpendingfootnotes{\LWR@depthsection}%
5430 \LWR@stoppars%
5431
5432 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
5433
5434 \LWR@section{#1}{#2}{#4}{section}%
5435 }

```

`\subsection * [⟨TOC name⟩] {⟨name⟩}`

```

5436 \DeclareDocumentCommand{\subsection}{s o m}{%
5437 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsection}%
5438 \LWR@stoppars%

```

```

5439
5440 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
5441
5442 \LWR@section{#1}{#2}{#3}{subsection}%
5443 }

```

`\subsubsection` * [*TOC name*] {*name*}

```

5444 \DeclareDocumentCommand{\subsubsection}{s o m}{%
5445 \LWR@maybeprintpendingfootnotes{\LWR@depthsubsubsection}%
5446 \LWR@stoppars%
5447
5448 \LWR@startnewdepth{\LWR@depthsubsubsection}%
5449 {\LWR@printclosesubsubsection}%
5450
5451 \LWR@section{#1}{#2}{#3}{subsubsection}%
5452 }

```

`\paragraph` * [*TOC name*] {*name*}

```

5453 \DeclareDocumentCommand{\paragraph}{s o m}{%
5454 \LWR@maybeprintpendingfootnotes{\LWR@depthparagraph}%
5455 \LWR@stoppars%
5456
5457 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
5458
5459 \LWR@section{#1}{#2}{#3}{paragraph}%
5460 }

```

`\subparagraph` * [*TOC name*] {*name*}

```

5461 \DeclareDocumentCommand{\subparagraph}{s o m}{%
5462 \LWR@maybeprintpendingfootnotes{\LWR@depthsubparagraph}%
5463 \LWR@stoppars%
5464
5465 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
5466
5467 \LWR@section{#1}{#2}{#3}{subparagraph}%
5468 }

5469 \end{warpHTML}

```

59 Starting a new file

for HTML & PRINT: 5470 \begin{warpall}

\HTMLLanguage Default language for the HTML lang tag.

```
5471 \newcommand*{\LWR@currentHTMLLanguage}{en-US}
5472
5473 \newcommand*{\HTMLLanguage}[1]{%
5474 \renewcommand*{\LWR@currentHTMLLanguage}{#1}%
5475 }

5476 \end{warpall}
```

for HTML output: 5477 \begin{warpHTML}

\LWR@filestart {*<title_suffix>*}

Creates the opening HTML tags.

```
5478 \newcommand*{\LWR@filestart}[1]{
5479 \LWR@traceinfo{\LWR@filestart !#1!}
```

Locally temporarily disable direct-formatting commands:

```
5480 \begingroup
5481 \LWR@nullfonts
```

Create the page's HTML header:

```
5482 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
```

The language is user-adjustable:

```
5483 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
5484 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
5485 \LWR@htmltag{meta charset="UTF-8" /\LWR@orignewline
```

Author:

```
5486 \ifthenelse{\equal{\theHTMLAuthor}{}}{%
5487 {}%
5488 {\LWR@htmltag{meta name="author" content="\theHTMLAuthor" /\LWR@orignewline}}%
```

```

5489 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%
5490 \LWR@originewline

```

```

5491 \ifdefempty{\LWR@currentHTMLDescription}{-}{%
5492 \LWR@htmltag{%
5493 meta name="description" content="\LWR@currentHTMLDescription" /}%
5494 \LWR@orignewline
5495 }%

```

```
5496 \LWR@htmltag{meta name="viewport" %
5497 content="width=device-width, initial-scale=1.0" /}%
5498 \LWR@orignewline
```

```
5499 \LWR<htmltag[!{-}{-}[if lt IE 9]]\LWR@orignewline
5500 \LWR<htmltag{%
5501 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"%}
5502 \LWR<htmltag[script]\LWR@orignewline
5503 \LWR<htmltag[!endif]{-}{-}\LWR@orignewline
```

```

5504 \ifthenelse{\equal{\theHTMLTitle}{}}%
5505 {}%
5506 {\LWR@htmltag{title}\theHTMLTitle#1\LWR@htmltag{/title}\LWR@orignewline}%

```

```
5507 \LWR@htmltag{%
5508 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
5509 \LWR@orignewline
```

```

5510 \ifbool{mathjax}%
5511 {%
5512     \begingroup%
5513     \LWR@restoreoriglists%
5514     \boolfalse{LWR@verbtags}

```

```

5515 \verbatiminput{lwarp_mathjax.txt}%
5516 \booltrue{LWR@verbtags}
5517 \endgroup%
5518 \LWR@stoppars
5519 }% end of mathjax
5520 {}%

```

End of the header:

```
5521 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```

5522 \LWR@htmltag{body}\LWR@orignewline
5523 \endgroup
5524 \LWR@traceinfo{LWR@filestart: done}
5525 }

5526 \end{warpHTML}

```

60 Starting HTML output

for HTML output: 5527 \begin{warpHTML}

`\LWR@LwarpStart` Executed at the beginning of the entire document.

```

5528 \catcode'\$=\active
5529 \newcommand*{\LWR@LwarpStart}
5530 {%
5531 \LWR@traceinfo{LWR@lwarpStart}

```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```

5532 \ifbool{FormatWP}{%
5533 \setcounter{FileDepth}{-5}%
5534 \boolfalse{HTMLDebugComments}%
5535 }{}

```

Expand and detokenize `\HomeHTMLFilename` and `\HTMLFilename`:

```

5536 \edef\LWR@strresult{\HomeHTMLFilename}
5537 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}
5538 \edef\LWR@strresult{\HTMLFilename}
5539 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}

```

Force onecolumn and empty page style:

```
5540 \LWR@origonecolumn%
5541 \LWR@origpagestyle{empty}%
```

No black box for overfull lines:

```
5542 \overfullrule=0pt
```

Reduce chance of line overflow in verbatim environments:

```
5543 \LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
5544 \LWR@origraggedright%
5545 \LetLtxMacro{\}\{\LWR@endoffline}%
```

Spread the lines for **pdftotext** to read them well:

```
5546 \linespread{1.3}%
```

For **pdftotext** to reliably identify paragraph splits:

```
5547 \setlength{\parindent}{0pt}
5548 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
5549 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space around the caption in the HTML:

```
5550 \setlength{\belowcaptionskip}{0ex}
5551 \setlength{\abovecaptionskip}{0ex}
```

Redefine the plain page style to be empty when used by index pages:

```
5552 \renewcommand{\ps@plain}{}%
```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```
5553 \LetLtxMacro\LWR@origtabular\tabular
5554 \LetLtxMacro\LWR@origendtabular\endtabular
```

```
5555 \LetLtxMacro\tabular\LWR@tabular
5556 \LetLtxMacro\endtabular\endLWR@tabular
```

Float captions:

```
5557 \let\LWR@origcaption\caption
```

Labels: `\ltx@label` is used in **amsmath** environments and is also patched by **cleveref**.

Label in HTML

```
5558 \let\LWR@origltx@label\ltx@label
5559 \let\ltx@label\LWR@htmlmathlabel
```

Logos:

```
5560 \let\TeX\LWR@TeX
5561 \let\LaTeX\LWR@LaTeX
5562 \let\LuaTeX\LWR@LuaTeX
5563 \let\LuaLaTeX\LWR@LuaLaTeX
5564 \let\XeTeX\LWR@XeTeX
5565 \let\XeLaTeX\LWR@XeLaTeX
5566 \let\ConTeXt\LWR@ConTeXt
```

Not yet started any paragraph handling:

```
5567 \global\boolfalse{LWR@doingapar}
5568 \global\boolfalse{LWR@doingstartpars}
```

Document and page settings:

```
5569 \mainmatter
5570 \LWR@origpagenumbering{arabic}
```

Start a new HTML file and a header:

```
5571 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
5572 \LWR@filestart{}
5573 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}
5574 \LWR@htmltag{header}\LWR@orignewline
5575 \LWR@startpars
5576 \LWR@firstpagetop
5577 \LWR@stoppars
5578 \LWR@htmltag{/header}\LWR@orignewline
5579 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
5580 \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`.

```
5581 \LWR@patchlists
```

Ensure that math mode is active to call **lwarp**'s patches:

```
5582 \catcode'\$=\active
```

Required for `\nameref` to work with svg math:

```
5583 \immediate\write\@mainaux{\catcode'\string$\active}%
5584 \LetLtxMacro\LWR@syntaxhighlightone$% balance for editor syntax highlighting
```

Allow HTML paragraphs to begin:

```
5585 \LWR@startpars
```

If using MathJax, disable `\ensuremath` by printing a nullified definition at the start of each file, and add further customizations:

```
5586 \LWR@customizeMathJax

5587 \LWR@traceinfo{\LWR@lwarpStart: done}
5588 }
5589 \catcode'\$=3% math shift until lwarp starts

5590 \end{warpHTML}
```

61 Ending HTML output

for HTML output: `5591 \begin{warpHTML}`

`\LWR@requesttoc` $\{\langle\textit{boolean}\rangle\}$ $\{\langle\textit{suffix}\rangle\}$ Requests that a toc, lof, or lot be generated.

```
5592 \newcommand*\LWR@requesttoc[2]{%
5593 \ifbool{#1}
5594 {
5595   \expandafter\newwrite\csuse{tf@#2}
5596   \immediate\openout \csuse{tf@#2} \jobname.#2\relax
5597 }{}
5598 }
```

\LWR@LwarpEnd Final stop of all HTML output:

```
5599 \newcommand*{\LWR@LwarpEnd}
5600 {
5601 \LWR@stoppars
5602 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
5603 \LWR@htmlelementclassend{section}{textbody}
```

Print any pending footnotes:

```
5604 \LWR@printpendingfootnotes
```

Create the footer:

```
5605 \LWR@htmlelement{footer}
5606
5607 \LWR@pagebottom
5608
5609 \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.

```
5610 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWP}}
5611 {}
5612 {
5613   \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
5614 }
```

```
5615 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
5616 \LWR@htmltag{/body}\LWR@orignewline
5617 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
5618 \LWR@orignewpage
```

For lateximage commands:

```
5619 \immediate\closeout\LWR@lateximagesfile
5620 }
```

```
5621 \end{warpHTML}
```

62 Title page

package support



load order

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

`\published` and `\subtitle`

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

affiliation

lwarp provides for the `\author` macro an additional `\affiliation` macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using **titlingpage**'s `\theauthor` in the main text.

reusing titlepage information

The **titling** package maintains the definitions of `\thetitle`, `\theauthor`, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the `\thanks` and `\affiliation` removed, and for `\author` the `\and` is replaced to generate a simple inline list of authors separated by commas. Note: `\theauthor` does not work well with **authblk** unless the traditional \TeX syntax is used.



`\theauthor`, **authblk**

custom titlepages

`\printtitle`, `\printauthor`, etc., are provided for use inside a custom titlepage or **titlingpage** environment, and these retain the `\thanks` and `\affiliation`.

`\printthanks`

`\printthanks` has been added to force the printing of thanks inside a **titlingpage** environment when `\maketitle` is not used.



Inside a `\titlepage` or `\titlingpage` environment, use `\thanks` instead of `\footnote` for acknowledgements, etc.

62.1 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-

`\and`

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` `\@author` `\@date` `\@title`, `\@author`, etc. store the values as originally assigned, including any `\thanks`, `\and`, or `\affiliation`. These are low-level macros intended to be used by other macros only inside a `titlepage` or `titlingpage`, and are used by `\maketitle`. The author is printed inside a single-column tabular, which becomes multiple single-column tabulars if multiples authors are included. For HTML these tabulars become side-by-side <div>s of class `oneauthor`, all of which are combined into one <div> of class `author`.

`\printtitle` `\printauthor` `\printdate` `\printtitle`, etc. are user-level macros intended to be used in custom `titlepage` or `titlingpage` environments in cases where `\maketitle` is not desired. These commands preserve the `\thanks`, etc., and should not be used in the main text.

`\thetitle` `\theauthor` `\thedata` `\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 \LaTeX syntax for `\author`, optionally with **lwarp**'s `\affiliation` macro as well.

⚠ **affiliations** After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

\affiliation A solution, provide here, is to define a macro `\affiliation` which, during `\maketitle`, starts a new row and adds the affiliation, but after `\maketitle` is finished `\affiliation`

is re-defined to discard its argument, thus printing only the author names when `\author` is later used inline.

62.2 `\if@titlepage`

for HTML & PRINT: 5622 `\begin{warpall}`

`\if@titlepage` Some classes do not provide `\if@titlepage`. In this case, provide it and force it false.

```
5623 \ifcsvoid{@titlepagefalse}{
5624     \newif\if@titlepage
5625     \@titlepagefalse
5626 }{}

5627 \end{warpall}
```

62.3 Changes for `\affiliation`

`\affiliation` `{<text>}`

Adds the affiliation to the author for use in `\maketitle`.

Inside `titlepage`, this macro prints its argument. Outside, it is null.

for HTML & PRINT: 5628 `\begin{warpall}`
 5629 `\providerobustcmd{\affiliation}[1]{}%`
 5630 `\end{warpall}`

for PRINT output: 5631 `\begin{warpprint}`

```
5632 \AtBeginEnvironment{titlepage}{
5633 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
5634 }
5635
5636 \AtBeginDocument{
5637 \ifpackageloaded{titling}{
5638 \AtBeginEnvironment{titlingpage}{
5639 \renewrobustcmd{\affiliation}[1]{\ \textsc{\small#1}}
5640 }
5641 }{}% titling loaded
5642 }% AtBeginDocument

5643 \end{warpprint}
```

for HTML output: 5644 \begin{warpHTML}

Env titlepage Sets up a <div> of class titlepage. Provided even for **memoir** class, since it is used by \maketitle.

```
5645 \DeclareDocumentEnvironment{titlepage}{}
5646 {
5647 \renewrobustcmd{\affiliation}[1]{\ \ \InlineClass{affiliation}{##1}}
5648 \LWR@printpendingfootnotes
5649 \LWR@forcenewpage
5650 \BlockClass{titlepage}
5651 }
5652 {
5653 \endBlockClass
5654 \LWR@printpendingfootnotes
5655 }

5656 \end{warpHTML}
```

62.4 Printing the thanks

for HTML & PRINT: 5657 \begin{warpall}

\printthanks Forces the \thanks to be printed.

This is necessary in a titlingpage environment when \maketitle was not used.

```
5658 \newcommand*{\printthanks}{\@thanks}

5659 \end{warpall}
```

62.5 Printing the title, etc. in HTML

The following are for printing the title, etc. in a titlepage or a titlingpage in HTML:

for HTML output: 5660 \begin{warpHTML}

\printtitle

```
5661 \newcommand*{\printtitle}
5662 {
5663 \LWR@stoppars
```

```

5664 \LWR@htmltag{\LWR@tagtitle}%
5665 \@title%
5666 \LWR@htmltag{\LWR@tagtitleend}
5667 \LWR@startpars
5668 }

```

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

```

5669 \newcommand*{\LWR@printthetitle}
5670 {
5671 \LWR@stoppars
5672 \LWR@htmltag{\LWR@tagtitle}%
5673 \thetitle%
5674 \LWR@htmltag{\LWR@tagtitleend}
5675 \LWR@startpars
5676 }

```

`\printauthor` HTML version.

```

5677 \newcommand*{\printauthor}{

```

The entire author block is contained in a <div> named author:

```

5678 \begin{BlockClass}{author}

```

`\and` finishes one author and starts the next:

```

5679 \renewcommand{\and}{%
5680 \end{BlockClass}
5681 \begin{BlockClass}{oneauthor}
5682 }

```

Individual authors are contained in a <div> named oneauthor:

```

5683 \begin{BlockClass}{oneauthor}
5684 \@author
5685 \end{BlockClass}
5686 \end{BlockClass}
5687 }

```

`\printdate`

```

5688 \newcommand*{\printdate}{%
5689 \begin{BlockClass}{titledate}
5690 \@date

```

```
5691 \end{BlockClass}
5692 }
```

```
5693 \end{warpHTML}
```

62.6 Printing the title, etc. in print form

The following are for printing the title, etc. in a titlepage or a titlingpage in print form:

for PRINT output: 5694 \begin{warpprint}

\printtitle

```
5695 \newcommand*{\printtitle}{\Huge\@title}}
```

\printauthor Print mode.

```
5696 \newcommand*{\printauthor}
```

```
5697    {{\large\begin{tabular}{t}{c}\@author\end{tabular}}}
```

\printdate

```
5698 \newcommand*{\printdate}{\small\textit{\@date}}}
```

```
5699 \end{warpprint}
```

62.7 \maketitle for HTML output

An HTML <div> of class titlepage is used.

\thanks are a form of footnotes used in the title page. See section 55 for other kinds of footnotes.

See \thanksmarkseries{series}, below, to set the style of the footnote marks.

for HTML output: 5700 \begin{warpHTML}

```
5701 \@ifclassloaded{memoir}
```

```
5702 {
```

```
5703 \newcommand{\LWR@setfootnoteseries}{%
```

```
5704    \renewcommand\thefootnote{\@arabic\c@footnote}%
```

```

5705 }
5706 }{% not memoir
5707 \if@titlepage
5708 \newcommand{\LWR@setfootnoteseries}{%
5709     \renewcommand\thefootnote{\@arabic\c@footnote}%
5710 }
5711 \else
5712 \newcommand{\LWR@setfootnoteseries}{%
5713     \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
5714 }
5715 \fi
5716 }{% not memoir

```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```

5717 \newcommand*{\LWR@maketitlesetup}{%

```

Redefine the footnote mark:

```

5718 \LWR@setfootnoteseries%
5719 \def\@makefnmark{\textsuperscript{\thefootnote}}

```

```

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}

```

Redefine the footnote text:

```

5720 \long\def\@makefntext##1{%

```

Make the footnote mark and some extra horizontal space for the tags:

```

5721 \textsuperscript{\@thefnmark}~%

```

```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)

```

Print the text:

```

5722 ##1%
5723 }%
5724 }

```

`\@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**.

```
5725 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
5726   \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
5727   **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
5728   \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.

```
5729 \renewcommand*{\maketitle}{%
```

An HTML titlepage `<div>` is used for all classes.

```
5730 \begin{titlepage}
```

Set up special patches:

```
5731 \LWR@maketitlesetup
```

Typeset the title, etc:

```
5732 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
5733 \@thanks
```

Close the HTML titlepage div and cleanup:

```
5734 \end{titlepage}
5735 \setcounter{footnote}{0}%
5736 \global\let\thanks\relax
5737 \global\let\maketitle\relax
5738 \global\let\@maketitle\relax
5739 \global\let\@thanks\@empty
5740 \global\let\@author\@empty
5741 \global\let\@date\@empty
5742 \global\let\@title\@empty
5743 \global\let\title\relax
5744 \global\let\author\relax
5745 \global\let\date\relax
5746 \global\let\and\relax
5747 }
```

`\@maketitle` HTML mode. Typesets the title, etc.:

```
5748 \DeclareDocumentCommand{\@maketitle}{\}%
5749   \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
5750   \@title
5751   \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
5752   \begin{BlockClass}{author}
```

For **IEEEtran** class:

```
5753   \renewcommand*{\cr}{\}
5754   \renewcommand*{\crr}{\}
5755   \renewcommand*{\noalign}{\}

5756   \renewcommand{\and}{\{
5757     \end{BlockClass}
5758     \begin{BlockClass}{oneauthor}
5759   \}
5760   \begin{BlockClass}{oneauthor}
5761     \@author
5762   \end{BlockClass}
5763 \end{BlockClass}
5764 \begin{BlockClass}{titledate}
5765   \@date
5766 \end{BlockClass}
5767 }
```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```
5768 \newcommand*{\LWR@titlingmaketitle}{\%
```

Keep pending footnotes out of the title block:

```
5769 \@thanks
```

Set up special patches:

```
5770 \LWR@maketitlesetup
```

Typeset the title, etc:

```
5771 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
5772 \@thanks
5773 }
```


```
5774 \end{warpHTML}
```

62.8 \published and \subtitle

\subtitle and \published To add \subtitle and \published to the titlepage, load the **titling** package and use \AddSubtitlePublished in the preamble.

The default lwarp.css has definitions for the published and subtitle classes.

If **titling** is loaded, \AddSubtitlePublished creates a number of additional macros, and also assigns some of the **titling** hooks. If **titling** is not loaded, \AddSubtitlePublished creates null macros.

 **titling hooks** Do not use \AddSubtitlePublished if the user has patched the **titling** hooks for some other reason. Portions are marked \warpprintonly to reduce extra tags in HTML. Similarly, BlockClass has no effect in print mode. Thus, the following may be marked warpall.

for HTML & PRINT: 5775 \begin{warpall}

\AddSubtitlePublished Adds \published and \subtitle, and related.

```
5776 \newcommand*{\AddSubtitlePublished}{%
5777 \ifpackageloaded{titling}{% yes titling package
5778   \newcommand{\@published}{}%
5779   \newcommand{\published}[1]{\gdef\@published{##1}}%
5780   \renewcommand*{\maketitlehooka}{\printpublished}%
5781   \newcommand*{\printpublished}{%
5782     \warpprintonly{\begin{center}\unskip}%
5783     \begin{BlockClass}{published}%
5784     \warpprintonly{\large\itshape}%
5785     \@published%
5786     \end{BlockClass}%
5787     \warpprintonly{\end{center}}}%
5788   }%
5789   \newcommand{\@subtitle}{}%
5790   \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}%
5791   \renewcommand*{\maketitlehookb}{\printsubtitle}%
5792   \newcommand*{\printsubtitle}{%
5793     \warpprintonly{\begin{center}\unskip}%
5794     \begin{BlockClass}{subtitle}%
5795     \warpprintonly{\Large\itshape}%
5796     \@subtitle%
5797     \end{BlockClass}%
5798     \warpprintonly{\end{center}}}%
5799   }%
```

```

5800 }% yes titling package
5801 {% no titling package
5802     \newcommand{\published}[1]{}%
5803     \newcommand*{\printpublished}{}%
5804     \newcommand{\subtitle}{}%
5805     \newcommand*{\printssubtitle}%
5806 }% no titling package
5807 }% \AddSubtitlePublished

5808 \end{warpall}

```

63 Abstract

The following code replaces the \TeX default, and will itself be replaced later if the **abstract** package is loaded.

for HTML output: 5809 \begin{warpHTML}

\abstractname User-redefinable title for the abstract.

Also over-written by the **babel** package.

```
5810 \providecommand*{\abstractname}{Abstract}
```

Some classes allow an optional name, so it is allowed here.

Env **abstract**

```

5811 \DeclareDocumentEnvironment{abstract}{0{\abstractname}}
5812 {
5813 \LWR@forcenewpage
5814 \BlockClass{abstract}
5815 \BlockClassSingle{abstracttitle}{#1}
5816 }
5817 {
5818 \endBlockClass
5819 }

5820 \end{warpHTML}

```

64 Quote and verse

64.1 Attributions

`\attribution` For use with quote, quotation, verse:

Ex: "A quotation." `\attribution{\textsc{Author Name}\\\textsl{Book Title}}`

for HTML output:

```
5821 \begin{warpHTML}
5822 \newcommand{\attribution}[1]{%
5823     \begin{BlockClass}{attribution}
5824     #1
5825     \end{BlockClass}
5826 }
5827 \end{warpHTML}
```

for PRINT output:

```
5828 \begin{warpprint}
5829 \newcommand{\attribution}[1]{
5830     \begin{flushright}
5831     \unskip
5832     #1
5833     \end{flushright}%
5834 }
5835 \end{warpprint}
```

64.2 Quotes, quotations

for HTML output:

```
5836 \begin{warpHTML}
```

Env `quote`

```
5837 \renewenvironment*{quote}
5838 {
5839 \LWR@forcenewpage
5840 \LWR@htmlblocktag{blockquote}
5841 }
5842 {\LWR@htmlblocktag{/blockquote}}
5843
5844 \renewenvironment*{quotation}
5845 {
5846 \LWR@forcenewpage
5847 \LWR@htmlblocktag{blockquotation}
5848 }
5849 {\LWR@htmlblocktag{/blockquotation}}
```

```
5850 \end{warpHTML}
```

64.3 Verse

When using **verse** or **memoir**, always place a `\\` after each line.


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

```
\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}
```

<p>Len <code>\leftskip</code></p> <p>Len <code>\leftmargini</code></p> <p>Len <code>\HTMLleftskip</code></p> <p>Len <code>\HTMLleftmargini</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>\HTMLleftskip</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.

64.3.1 \LaTeX core verse environment

for HTML output: 5851 `\begin{warpHTML}`

Env **verse**

```
5852 \renewenvironment{verse}
5853     {\let\\ \newline% lwarp
5854      \list{}{\itemsep      \z@
5855              \itemindent   -1.5em%
5856              \listparindent\itemindent}
```

```

5857             \rightmargin \leftmargin
5858             \advance\leftmargin 1.5em}%
5859         \item\relax}
5860     {\endlist}

5861 \end{warpHTML}

```

for HTML & PRINT: 5862 \begin{warpall}

64.3.2 verse and memoir

The following lengths are used by **verse** and **memoir**. They may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

Len `\HTMLvleftskip` Sets `\vleftskip` inside a verse environment in HTML.

```

5863 \newlength{\HTMLvleftskip}
5864 \setlength{\HTMLvleftskip}{1em}

```

Len `\HTMLleftmargini` Sets `\leftmargini` inside a verse environment in HTML.

```

5865 \newlength{\HTMLleftmargini}
5866 \setlength{\HTMLleftmargini}{4.5em}

```

```

5867 \end{warpall}

```

65 Verbatim and tabbing

for HTML & PRINT: 5868 \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.

```

5869 \newlength{\VerbatimHTMLWidth}
5870 \setlength{\VerbatimHTMLWidth}{4in}
5871 \end{warpall}

```

for HTML output: 5872 \begin{warpHTML}

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML

head.

```
5873 \newbool{LWR@verbtags}
5874 \booltrue{LWR@verbtags}
```

```
\LWR@atbeginverbatim [\langle 1: style \rangle] {\langle 2: negative \baselineskip \vspace \rangle} {\langle 3: class \rangle}
```

Encloses a verbatim environment with the given css class.

```
5875 \newcommand*{\LWR@atbeginverbatim}[3] []
5876 {%
```

Avoid excessive space between lines:

```
5877 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
5878 \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.

```
5879 \ifbool{LWR@verbtags}{%
5880   \LWR@htmltag{pre class="#3"
5881     \ifthenelse{\equal{#1}{}}{}{style="#1"}}%
5882   }%
5883   \LWR@orignewline% pre
5884   \leavevmode\unskip\LWR@print@vspace*{-#2\baselineskip}%
5885 }{}%
```

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.

```
5886 \begingroup%
```

```
5887 % \LWR@orignormalsize%
5888 \LWR@origttfamily%
5889 \LWR@origsmall%
```

Since inside a <pre>, restore the original list processing:

```
5890 \LWR@restoreoriglists%
```

Turn off **babel-french** extra space before punctuation:

```
5891 \LWR@FBcancel%
```

Do not produce HTML tags for \hspace inside a verse par. Restore plain \TeX \hspace functionality:

```
5892 \LWR@select@print@hspace%
5893 }
```

`\LWR@afterendverbatim` $\{ \langle negative \ baselineskip \ vspace \rangle \}$

Finishes enclosing a verbatim environment.

```
5894 \newcommand*\LWR@afterendverbatim}[1]{%
5895 \endgroup%
5896 \par%
```

At the end of the environment, close the pre:

```
5897 \ifbool\LWR@verbtags){%
5898     \LWR@print@vspace*{-#1\baselineskip}%
5899     \noindent\LWR@htmltag{/pre}\LWR@orignewline% pre
5900 }{}%
```

Resume regular paragraph handling:

```
5901 \LWR@startpars%
5902 }
```

`\verbatiminput` $\{ \langle filename \rangle \}$

Patch \verbatiminput to add HTML tags:

```
5903 \let\LWRV@origverbatim@input\verbatim@input
5904
5905 \renewcommand\verbatim@input[2]{%
5906 \ifbool\LWR@verbtags){\LWR@forcenewpage}{}%
5907 \LWR@atbeginverbatim{2.5}{Verbatim}%
5908 \LWRV@origverbatim@input{#1}{#2}%
5909 \LWR@afterendverbatim{1.5}%
5910 }
```

Env `verbatim`

```
5911 \AfterEndPreamble{
5912 \LWR@traceinfo{Patching verbatim.}
5913 \AtBeginEnvironment{verbatim}{%
5914 \LWR@forcenewpage%
5915 \LWR@atbeginverbatim{2.5}{verbatim}%
5916 }
```

```

5917 \AfterEndEnvironment{verbatim}{%
5918     \LWR@afterendverbatim{1}%
5919 }
5920 }

```

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

[math in tabbing](#) If `math` is used inside `tabbing`, place `tabbing` inside a `lateximage` environment, which will render the entire environment as a single `svg` image.

```

5921 \LetLtxMacro\LWR@origtabbing\tabbing
5922 \LetLtxMacro\LWR@origendtabbing\endtabbing
5923
5924 \renewcommand*{\tabbing}{%
5925     \LWR@forcenewpage%
5926     \LWR@atbeginverbatim{3}{tabbing}%
5927     \LWR@origtabbing%
5928 }
5929
5930 \renewcommand*{\endtabbing}{%
5931     \LWR@origendtabbing%
5932     \LWR@afterendverbatim{1}%
5933 }

```

`\AtBeginDocument` because `\LWR@restoreorigformatting` has not yet been defined:

```

5934 \AtBeginDocument{
5935     \appto\LWR@restoreorigformatting{%
5936         \LetLtxMacro\tabbing\LWR@origtabbing%
5937         \LetLtxMacro\endtabbing\LWR@origendtabbing%
5938     }
5939 }

5940 \end{warpHTML}

```

66 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: 5941 `\begin{warpHTML}`

`\@begintheorem` `{\langle name \rangle}` `{\langle number \rangle}`

```
5942 \renewcommand{\@begintheorem}[2]{%
5943 \LWR@forcenewpage
5944 \BlockClass{theoremcontents}
5945 \trivlist
5946 \item[\InlineClass{theoremlabel}{#1\ #2\ }]\itshape
5947 }
```

`\@opargbegintheorem` `{\langle name \rangle}` `{\langle number \rangle}` `{\langle oparg \rangle}`

```
5948 \renewcommand{\@opargbegintheorem}[3]{%
5949 \LWR@forcenewpage
5950 \BlockClass{theoremcontents}
5951 \trivlist
5952 \item[\InlineClass{theoremlabel}{#1\ #2\ (#3)\ }]\itshape
5953 }
```

`\@endtheorem`

```
5954 \renewcommand*\@endtheorem{%
5955 \endtrivlist
5956 \endBlockClass% theoremcontents
5957 }
```

```
5958 \end{warpHTML}
```

67 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 T_EX horizontal skip and fill commands, allowing the user to define `\makelabel` for print mode while HTML mode ignores those commands.

 **label font** When defining `\makelabel` in a list environment, use `\textbf` etc. instead of `\bfseries`.

67.1 List environment

for HTML output: 5959 `\begin{warpHTML}`

`\LWR@printcloselist` May be locally redefined by `enumerate` or `description`.

```
5960 \newcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
```

`\LWR@printopenlist` May be locally redefined by `enumerate` or `description`.

```
5961 \newcommand*{\LWR@printopenlist}{ul style="\LWR@print@mbox{list-style-type:none}"}
```

`\@mklab` Removes PDF spacing.

```
5962 \AtBeginDocument{
5963 \def\@mklab#1{%
5964 %     \hfil %
5965     #1}
5966 \let\makelabel\@mklab
5967 }
```

`\@donoparitem` Modified for HTML output by replacing \TeX boxes with plain text. Also removes PDF spacing.

```

5968 \def\@donoparitem{%
5969   \@noparitemfalse
5970 %   \global\setbox\@labels\hbox{\hskip -\leftmargin
5971 %                               \unhbox\@labels
5972 %                               \hskip \leftmargin}%
5973 %   \if@minipage\else
5974 %     \@tempskipa\lastskip
5975 %     \vskip -\lastskip
5976 %     \advance\@tempskipa\@outerparskip
5977 %     \advance\@tempskipa -\parskip
5978 %     \vskip\@tempskipa
5979 %   \fi
5980 }
```

`\@item` Modified for HTML output by replacing \TeX boxes with plain text. Also removes PDF spacing.

```

5981 \def\LWR@HTML@item[#1]{%
5982 \LWR@traceinfo{\@item}
5983   \if@noparitem
5984     \@donoparitem
5985   \else
5986 %     \if@inlabel
5987 %       \indent
5988 %       \par
5989 %     \fi
5990     \ifhmode
5991 %       \unskip\unskip
5992 %       \par
5993 %     \fi
5994     \if@newlist
5995       \if@nobreak
5996         \@nbitem
5997       \else
5998 %         \addpenalty\@beginparpenalty
5999 %         \addvspace\@topsep
6000 %         \addvspace{-\parskip}%
6001 %       \fi
6002     \else
6003 %       \addpenalty\@itempenalty
6004 %       \addvspace\itemsep
6005 %     \fi
6006     \global\@inlabeltrue
6007   \fi
6008 %   \everypar{%
6009 %     \@minipagefalse
```

```

6010    \global\@newlistfalse

6011 %    \if@inlabel
6012 %    \global\@inlabelfalse

6013 %    {\setbox\z@\lastbox
6014 %    \ifvoid\z@
6015 %    \kern-\itemindent
6016 %    \fi}%

6017 %    \box\@labels
6018 %    \penalty\z@
6019 %    \fi

6020 %    \if@nobreak
6021 %    \@nobreakfalse
6022 %    \clubpenalty \@M
6023 %    \else
6024 %    \clubpenalty \@clubpenalty
6025 %    \everypar{}\%
6026 %    \fi}%

6027 \if@noitemarg
6028 \@noitemargfalse
6029 \if@nmbrlist

6030    \refstepcounter\@listctr
6031    \fi
6032 \fi

6033 \makelabel{#1} % extra space
6034 % \sbox\@tempboxa{\makelabel{#1}}%
6035 % \global\setbox\@labels\hbox{%
6036 % \unhbox\@labels
6037 % \hskip \itemindent
6038 % \hskip -\labelwidth
6039 % \hskip -\labelsep
6040 % \ifdim \wd\@tempboxa >\labelwidth
6041 % \box\@tempboxa

6042 % \else
6043 % \hbox to\labelwidth {\unhbox\@tempboxa}%
6044 % \fi
6045 % \hskip \labelsep}%
6046 \ignorespaces%
6047 }

```

`\@nbitem`

```
6048 \def\@nbitem{%
6049 %   \@tempskipa\@outerparskip
6050 %   \advance\@tempskipa -\parskip
6051 %   \addvspace\@tempskipa
6052 }
```

`\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.

```
6053 \newcommand*\LWR@listitem{%
6054 \LWR@stoppars%
6055 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
6056 \LWR@htmltag{li}%
6057 \LWR@startpars%
6058 \LWR@origitem%
6059 }
```

`\LWR@nulllistfills` Nullifies various \TeX fill commands, in case they are used inside `\makelabel`. Problems are caused when these are nullified all the time.

```
6060 \newcommand*\LWR@nulllistfills{%
6061 \renewcommand*\hss{}%
6062 \renewcommand*\llap[1]{##1}%
6063 \renewcommand*\rlap[1]{##1}%
6064 \renewcommand*\hfil{}%
6065 \renewcommand*\hfilneg{}%
6066 \renewcommand*\hfill{}%
6067 }
```

Env `list` {*<label>*} {*<commands>*}

```
6068 \newcommand*\LWR@liststart{%
6069 \LWR@traceinfo{\LWR@liststart}%
6070 \LWR@stoppars%
6071 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloselist}%
6072 \LWR@htmltag{\LWR@printopenlist}\LWR@originewline%
6073 \LWR@startpars%
6074 \setlength{\topsep}{0pt}%
6075 \setlength{\partopsep}{0pt}%
6076 \setlength{\itemsep}{0pt}%
6077 \setlength{\parsep}{0pt}%

```

```

6078 \setlength{\leftmargin}{0pt}%
6079 \setlength{\rightmargin}{0pt}%
6080 \setlength{\listparindent}{0pt}%
6081 \setlength{\itemindent}{0pt}%
6082 \setlength{\labelsep}{1em}%
6083 \LWR@nulllistfills%
6084 }

6085 \newcommand*{\LWR@listend}{%
6086 \LWR@traceinfo{\LWR@listend}%
6087 \LWR@stoppars%
6088 \LWR@closeprevious{\LWR@depthlist}%
6089 \LWR@startpars%
6090 }

```

67.2 Itemize

`\LWR@itemizeitem` [*<label>*]

Handles `\item` inside an itemize or enumerate.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```

6091 \newcommand*{\LWR@itemizeitem}{%
6092 \LWR@stoppars%
6093 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloselistitem}%
6094 \LWR@htmltag{li}%
6095 \LWR@startpars%
6096 \LWR@origitem%
6097 }

```

Env `itemize` [*<options>*]

```

6098 \newcommand*{\LWR@itemizestart}{%
6099 \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
6100 \renewcommand*{\LWR@printopenlist}{\ul style="\LWR@print@mbbox{list-style-type:none}}
6101 \let\item\LWR@itemizeitem%
6102 \LWR@nulllistfills%
6103 }

```

67.3 Enumerate

An HTML unordered list is used with customized \LaTeX -generated labels.

Env **enumerate** [*options*]

```
6104 \newcommand*{\LWR@enumeratestart}{%
6105 \renewcommand*{\LWR@printcloselist}{\LWR@printcloseitemize}
6106 \renewcommand*{\LWR@printopenlist}{\ul style="\LWR@print@mbx{list-style-type:none}"}
6107 \let\item\LWR@itemizeitem%
6108 \LWR@nulllistfills%
6109 }
```

67.4 Description

`\LWR@descitem` [*label*] Handles an `\item` inside a description.

```
6110 \newcommand*{\LWR@descitem}[1][]{%
6111 {%
6112 \LWR@stoppars%
6113 \LWR@setlatestname{#1}%
6114 \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.

```
6115 \LWR@select@html@nohspace%
```

Process the original `\item` code:

```
6116 \LWR@origitem[]%
```

Restore `\hspace` for use in the item text:

```
6117 \LWR@select@html@hspace%

6118 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
6119 \LWR@orignewline%
6120 \LWR@htmltag{dd}%
6121 \LWR@startpars%
6122 }
```

Env **description** [*options*]

```
6123 \newcommand*{\LWR@descriptionstart}{%
6124 \renewcommand*{\LWR@printcloselist}{\LWR@printclosedescription}
6125 \renewcommand*{\LWR@printopenlist}{\dl}
6126 \let\item\LWR@descitem%
6127 \LWR@nulllistfills%
6128 }
```

67.5 Patching the lists

`\LWR@patchlists` Patches list environments.

`\LWR@patchlists` remembers `\item` as defined by whatever packages have been loaded, then patches the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native \TeX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```

6129 \newcommand*\LWR@patchlists}{%
6130   \LetLtxMacro\item\LWR@listitem%
6131   \LetLtxMacro\@item\LWR@HTML@item%
6132   \renewcommand*\@trivlist}{%
6133     \LWR@traceinfo{@trivlist start}%
6134     \LWR@liststart%
6135     \LWR@orig@trivlist%
6136     \LWR@traceinfo{@trivlist done}%
6137   }%
6138   \renewcommand*\@trivlist}{%
6139     \LWR@traceinfo{trivlist}%
6140     \LWR@origtrivlist%
6141   }%
6142   \renewcommand*\@endtrivlist}{%
6143     \LWR@traceinfo{endtrivlist start}%
6144     \LWR@origendtrivlist\LWR@listend%
6145     \LWR@traceinfo{endtrivlist done}%
6146   }%
6147   \renewcommand*\@itemize}{%
6148     \LWR@itemizestart\LWR@origitemize%
6149   }%
6150   \renewcommand*\@enumerate}{%
6151     \LWR@enumeratestart\LWR@origenumerate%
6152   }%
6153   \renewcommand*\@description}{%
6154     \LWR@descriptionstart\LWR@origdescription%
6155   }%
6156 }
```

`\LWR@restoreoriglists` Restores the original `trivlist` environment.

```

6157 \newcommand*\LWR@restoreoriglists}{%
6158   \LWR@traceinfo{\LWR@restoreoriglists}%
6159   \LetLtxMacro\item\LWR@origitem%
6160   \LetLtxMacro\@item\LWR@orig@item%
6161   \let\@trivlist\LWR@orig@trivlist%
6162   \let\trivlist\LWR@origtrivlist%
6163   \let\endtrivlist\LWR@origendtrivlist%
6164   \LetLtxMacro\itemize\LWR@origitemize%
```

```

6165 \LetLtxMacro\enditemize\LWR@endorigitemize%
6166 \LetLtxMacro\enumerate\LWR@origenumerate%
6167 \LetLtxMacro\endenumerate\LWR@endorigenumerate%
6168 \LetLtxMacro\description\LWR@origdescription%
6169 \LetLtxMacro\enddescription\LWR@endorigdescription%
6170 \let\@mklab\LWR@orig@mklab%
6171 \let\makelabel\LWR@origmakelabel%
6172 \let\@donoparitem\LWR@orig@donoparitem%
6173 \let\@nbitem\LWR@orig@nbitem%
6174 }

6175 \end{warpHTML}

```

68 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax of the \TeX core and the various tabular-related packages.

68.1 Limitations

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, multirows, * column specifiers, **siunitx** S columns, or the packages **multirow**, **longtable**, **supertabular**, or **xtab**.

Defining environments:

⚠ misplaced alignment
alignment tab character &

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

```
\StartDefiningTabulars
```

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

```
\StopDefiningTabulars
```

⚠ floatrow

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

⚠ tabular inside another
environment

- When creating a new environment which contains a tabular environment, **lwarp**'s emulation of the tabular does not automatically resume when the containing 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
}
\StopDefiningTabulars

```

Cell contents:

⚠ paragraphs

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

⚠ `\multirow`

- For **multirow**, insert `\mrowcell` into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.

```

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

```

vposn

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

- The **multirow** documentation regarding colored cells recommends using a negative number of rows. This will not work with **lwarp**, so `\warpprintonly` and `\warphTMLonly` must be used to make versions for print and HTML.
- See section 260.2 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

- Doubled `\hlines`, `\midrules`, and vertical rules are supported.
- Vertical rules next to either side of an @ or ! column are displayed on both sides of the column.

width and trim

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

full-width rules

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

combined rules

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

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

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

⚠ \warpprintonly
misplaced \noalign

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

```
This & That \endhead
```

to

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

and likewise with the other `\end` headings. Keep the `\endfirsthead` row unchanged, as it is still relevant to HTML output.

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 \\}
```

for HTML output: 6176 `\begin{warppHTML}`

68.2 Temporary package-related macros

These macros are temporary placeholders for macros defined by various packages. If the relevant package is not loaded, these placeholders are used instead.

68.2.1 arydshln

Emulated by the original \TeX non-dashed versions.

```
6177 \LetLtxMacro\hdashline\hline
6178 \LetLtxMacro\cdashline\cline
6179 \LetLtxMacro\firsthdashline\hline
6180 \LetLtxMacro\lasthdashline\hline
```

68.3 Token lookahead

Used by `\LWR@futurenonSPACElet` to look at the next token.

\LWR@mynexttoken

```
6181 \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

```
6182 \def\LWR@futurenonSPACElet#1{\def\LWR@cs{#1}%
6183 \afterassignment\LWR@fnslone\let\nexttoken= }
6184 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
6185 \def\LWR@fnsltwo{%
6186 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
6187 \else\let\next=\nexttoken\fi\next}
6188 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
6189 \newcommand*\LWR@getmynexttoken}{%
6190 \LWR@traceinfo{\LWR@getmynexttoken}%
6191 % nothing must follow this next line
6192 \LWR@futurenonSPACElet\LWR@mynexttoken\LWR@tabledatacolumnntag
6193 }
```

68.4 Tabular variables

Bool LWR@startedrow True if should print a row tag before this column.

```
6194 \newbool{LWR@startedrow}
6195 \boolfalse{LWR@startedrow}
```

Bool LWR@tabularcelladded True if have added a data cell for this position.

```
6196 \newbool{LWR@tabularcelladded}
6197 \boolfalse{LWR@tabularcelladded}
```

Ctr LWR@hlines Number of \hlines or \midrules above the next row.

```
6198 \newcounter{LWR@hlines}
```

Ctrl	LWR@hdashedlines	Number of arydshln dashed lines above the next row.
		6199 \newcounter{LWR@hdashedlines}
Bool	LWR@doingtbrule	True if the next row will have a top/bottom rule above it.
		6200 \newbool{LWR@doingtbrule}
		6201 \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.
		6202 \newbool{LWR@doingcmidrule}
		6203 \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.
		6204 \newbool{LWR@tableparcell}
Bool	LWR@skippingmrowcell	True if are doing an empty \multirow cell, and thus there is no data tag to close.
		6205 \newbool{LWR@skippingmrowcell}
	Bool	True if are doing an empty \multicolumnrow cell, and thus there is no data tag to close, and do not print @ and ! columns.
LWR@skippingmcolrowcell		6206 \newbool{LWR@skippingmcolrowcell}
Bool	LWR@skipatbang	True if just finished a \multicolumn so should not create the trailing @ or ! columns table data cells.
		6207 \newbool{LWR@skipatbang}
Bool	LWR@emptyatbang	True if finishing a row and should print empty @ or ! column table data cells.
		6208 \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.
		6209 \newbool{LWR@intabularmetadata}
		6210 \boolfalse{LWR@intabularmetadata}
Ctrl	LWR@tabularDepth	Tracks whether & is being used inside a tabular.

```
6211 \newcounter{LWR@tabulardepth}
6212 \setcounter{LWR@tabulardepth}{0}
```

Ctrl 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.

```
6213 \newcounter{LWR@tabularpardepth}
6214 \setcounter{LWR@tabularpardepth}{0}
```

```
6215 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
6216 \newcommand*{\LWR@pposition}{}
6217 \newcommand*{\LWR@pleft}{}
6218 \newcommand*{\LWR@pright}{}%
```

\LWR@tablecolspec Holds the parsed column specification, of total width LWR@tabletotalcols, not counting @ and ! columns.

Will contain a string such as llrrccpc, exactly one letter per \LaTeX table column, without @, !, >, <, or the vertical bar.

```
6219 \newcommand*{\LWR@tablecolspec}{}%
```

\LWR@strresult Holds the result of Str functions.

```
6220 \providecommand*{\LWR@strresult}{}
6221 \providecommand*{\LWR@strresulttwo}{}%
```

\LWR@origcolspec Holds the original column specs given to tabular.

```
6222 \newcommand*{\LWR@origcolspec}{}%
```

Ctrl LWR@tablecolspecwidth Holds the number of tokens in the table columns specification.

This includes one for each @, !, <, > column, and also one for each of the parameters of p, @, !, <, > columns, and three for each D column.

(This is not the total # of \LaTeX columns in the table.)

```
6223 \newcounter{LWR@tablecolspecwidth}
```

Ctrl LWR@tablecolspecindex While parsing the \LaTeX table column specification, starts at 1 and is incremented per

token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
6224 \newcounter{LWR@tablecolspecindex}
```

Ctrl LWR@tablecolindex While parsing the \LaTeX table column specification, starts at 1 and is incremented per token of the specification. While producing the table, resets to 1 at the start of the table and also at each end of line, and is incremented by 1 by each ampersand.

```
6225 \newcounter{LWR@tablecolindex}
```

Ctrl LWR@tabletotalcols While parsing a table column specification, begins at 0 and increments by 1 per \LaTeX table column. Eventually holds the final number of \LaTeX table columns in each row, not counting @ and ! columns. (In HTML, @ and ! cells become their own columns, but are not included in LWR@tabletotalcols.)

```
6226 \newcounter{LWR@tabletotalcols}
```

Ctrl LWR@tabletotalcolsnext Holds the next \LaTeX table column index while parsing, equal to one more than LWR@tabletotalcols.

```
6227 \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 39.

LWR@colbangspec A data array of specifications for ! columns. The leftmost's index is leftedge, the others are counter values. See section 39.

LWR@colbeforespec A data array of specifications for > columns.

LWR@colafterspec A data array of specifications for < columns.

LWR@colbarspec A data array of specifications for vertical rules.

68.5 Handling &, @, !, and bar

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860>

```
\LWR@instertatbangcols
```

```
6228 \newcommand*{\LWR@insertatbangcols}{%
6229 \ifbool{LWR@skipatbang}%
```

```

6230 {}%
6231 {%
6232     \LWR@printatbang{at}{\arabic{LWR@tablecolindex}}%
6233     \LWR@printatbang{bang}{\arabic{LWR@tablecolindex}}%
6234 }%
6235 }

```

\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.

```

6236 \newcommand*{\LWR@closetabledatacell}{%
6237 \global\booltrue{LWR@intabularmetadata}%
6238 \ifbool{LWR@exitingtabular}{}%
6239 {% not exiting tabular
6240     \ifbool{LWR@skippingmrowcell} or bool{LWR@skippingmcolrowcell}}%
6241     {%

```

If not skipping a \multicolumnrow cell, insert the @ and ! columns after this non-existent column.

```

6242     \ifbool{LWR@skippingmcolrowcell}%
6243     {%
6244     {\LWR@insertatbangcols}%
6245     }%
6246     {% not skippingmrowcell

```

Insert any < then any @ and ! column contents, unless muted for the \bottomrule or a \multicolumn:

```

6247     \unskip%
6248     \ifbool{LWR@tabularmutemods} or
6249     bool{LWR@skipatbang} or
6250     bool{LWR@emptyatbang}
6251     }%
6252     {}%
6253     {\LWR@getexparray{LWR@colafterspec}{\arabic{LWR@tablecolindex}}}%
6254

```

Close paragraphs:

```

6255     \ifbool{LWR@tableparcel}{\LWR@stoppars}{}%
6256     \global\boolfalse{LWR@tableparcel}%

```

Close the table data cell.

Close any color <div>s.

```

6257      \whileboolexpr{test {\ifnumcomp{\value{LWR@cellcolordepth}}{>}{0}}}{%
6258          \LWR@htmltag{/div}\LWR@orignewline%
6259          \addtocounter{LWR@cellcolordepth}{-1}%
6260      }%

```

Skip the @ and ! cells if are closing a multicolumn cell.

```

6261      \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
6262      \global\booltrue{LWR@tabularcelladded}%
6263      \LWR@insertatbangcols%
6264      }% not skipping mrowcell
6265 }% not exiting tabular
6266 \global\boolfalse{LWR@skippingmrowcell}%
6267 \global\boolfalse{LWR@skippingmcolrowcell}%
6268 \global\boolfalse{LWR@skipatbang}%

```

Color control. Column is set by >{} for each cell, so it must be cleared here.

```

6269 \renewcommand*{\LWR@cellHTMLcolor}{}
6270 \renewcommand*{\LWR@columnHTMLcolor}{}
6271 \setcounter{LWR@cellcolordepth}{0}
6272 }

```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```

6273 \let\LWR@origampmacro&

6274 \end{warpHTML}

```

68.5.1 Handling &

for HTML output: 6275 \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.

```

6276 \newcommand*{\LWR@tabularampersand}{%
6277 \LWR@traceinfo{LWR@tabularampersand}%
6278 \ifnumcomp{\value{LWR@tabulardepth}}{>}{0}%
6279 {%

```

If not skipping a multirow cell, close the current data cell.

```
6280 \unskip%
6281 \LWR@closetabledatacell%
```

Move to the next column.

```
6282 \addtocounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
6283 \boolfalse{LWR@tabularcelladded}%
```

Look at the next token to decide multi or single column data tag.

```
6284 \LWR@getmynexttoken%
6285 }%
```

If not inside a tabular, performs the original action:

```
6286 {\LWR@origampmacro}%
6287 }
```

& is left with its original catcode for now.

tikz package seems to require & be left alone until after **tikz** has been loaded. Also, **cleveref** uses the ampersand in one of its options.

& is made active inside a tabular.

& is left alone when in math alignments.

68.5.2 Filling an unfinished row

`\LWR@tabularfinishrow` Adds empty table cells if necessary to finish the row.

At the end of the table, if any bottom rules are requested then an empty row must be generated to form the borders which show the rules.

```
6288 \newcommand*{\LWR@tabularfinishrow}{%
```

If not exiting the tabular, or doing a rule, or have already started a row, finish this row:

```
6289 \ifboolexpr{%
6290 not bool {LWR@exitingtabular} or%
```

```

6291     bool{LWR@doingtbrule} or%
6292     bool{LWR@doingcmidrule} or%
6293     test{\ifnumcomp{\value{LWR@hlines}}{>}{0}} or%
6294     test{\ifnumcomp{\value{LWR@hdashedlines}}{>}{0}} or%
6295     bool{LWR@startedrow}%
6296 }{%

```

To locally temporarily turn off LWR@exitingtabular so that table data tags will still be generated:

```

6297 \begingroup%

```

If generating a final row for the \bottomrule borders, turn off the @, !, <, and > column output:

```

6298 \ifbool{LWR@exitingtabular}{%
6299     \booltrue{LWR@tabularmutemods}%
6300 }{%

```

Reenable the table data tags until finished with the final row:

```

6301 \global\boolfalse{LWR@exitingtabular}%

```

Generate table data tags and ampersands until the right edge:

```

6302 \whileboolexpr{%
6303     test {
6304         \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
6305     } or %
6306     (%
6307         bool{LWR@intabularmetadata} and%
6308         not bool{LWR@tabularcelladded} and%
6309         test {
6310             \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}
6311         }%
6312     )%
6313 }%
6314 {%
6315     \LWR@tabledatasinglecolumnntag%

```

The following is essentially \LWR@tabularampersand with LWR@emptyatbang added to empty the following cells:

```

6316     \LWR@closetabledatacell%
6317     \addtocounter{LWR@tablecolindex}{1}%
6318     \boolfalse{LWR@tabularcelladded}%
6319     \global\booltrue{LWR@emptyatbang}%

```

Starts the next cell:

```
6320 \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
6321 {\LWR@getmynexttoken}%
6322 {}%
6323 }%
```

Reenable the original LWR@exitingtabular to close the entire table:

```
6324 \endgroup%
6325 \global\boolfalse{LWR@emptyatbang}%
6326 }{}% ifboolexpr
6327 }
```

68.6 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

\LWR@tabularendofline

```
6328 \NewDocumentCommand{\LWR@tabularendofline}{s o}{%
```

Finish the row:

```
6329 \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}%
6330 {\LWR@tabularfinishrow}%
6331 {\LWR@closetabledatacell}%
6332 \LWR@htmltag{/tr}\LWR@orignewline%
```

xcolor row color support:

```
6333 \@rowcolor%
```

No longer inside a data cell:

```
6334 \global\booltrue{LWR@intabularmetadata}%
```

Not yet started a table row:

```
6335 \global\boolfalse{LWR@startedrow}%
```

Additional setup:

```

6336 \setcounter{LWR@hlines}{0}%
6337 \setcounter{LWR@hdashedlines}{0}%
6338 \global\boolfalse{LWR@doingtbrule}%
6339 \global\boolfalse{LWR@doingcmidrule}%
6340 \LWR@clearmidrules%
6341 \renewcommand*{\LWR@rowHTMLcolor}{}%

```

Start at first column:

```

6342 \setcounter{LWR@tablecolindex}{1}%

```

Have not yet added data in this column:

```

6343 \boolfalse{LWR@tabularcelladded}%

```

Look at the next token to decide between single column data tag or a special case:

```

6344 \LWR@getmynexttoken%
6345 }

```

68.7 Looking ahead in the column specifications

`\LWR@columnspeclookahead` $\{ \langle offset \rangle \}$

Looks `offset` tokens ahead in the column specification, setting `\LWR@strresulttwo`.

The `w` column alignment will be seen as a single unit such as `{c}`.

```

6346 \newcommand*{\LWR@columnspeclookahead}[1]{%
6347 \setcounter{LWR@tempcountone}{\value{LWR@tablecolspecindex}}%
6348 \addtocounter{LWR@tempcountone}{#1}%
6349 \fullexpandarg%
6350 \StrChar{\LWR@origcolspec}{\arabic{LWR@tempcountone}}[\LWR@strresulttwo]%

```

Get the contents of the first group in `\LWR@strresulttwo`:

```

6351 \exploregroups%
6352 \StrChar{\LWR@strresulttwo}{1}[\LWR@strresulttwo]%
6353 \noexploregroups%
6354 }

```

68.8 Parsing @, >, <, !, bar columns

Holds the parsed argument for @, >, <, or ! columns:

```
6355 \newcommand*{\LWR@colparameter}{}
```

`\LWR@parseatcolumn` Handles @{text} columns.

```
6356 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
6357 \LWR@traceinfo{at column}%
6358 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into `\LWR@colparameter`, expanding once:

```
6359 \LWR@traceinfo{about to read the next token:}%
6360 \expandarg%
6361 \StrChar{\LWR@origcolspec}%
6362     {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
6363 \fullexpandarg%
```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```
6364 \LWR@traceinfo{have now read the next token}%
6365 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
6366 {% left edge of the table:
6367     \LWR@traceinfo{at the left edge}%
6368     \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
6369     \LWR@traceinfo{at the left edge: %
6370     \LWR@getexparray{LWR@colatspec}{leftedge}}%
6371 }%
6372 {% not at the left edge:
6373     \LWR@traceinfo{not at the left edge}%
6374     \LWR@setexparray{LWR@colatspec}%
6375         {\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
6376     \LWR@traceinfo{at \arabic{LWR@tabletotalcols}: %
6377     \LWR@getexparray{LWR@colatspec}{\arabic{LWR@tabletotalcols}}}%
6378 }%
6379 \let\LWR@colparameter\relax%
6380 \booltrue{LWR@validtablecol}%
6381 }
```

`\LWR@parsebangcolumn` Handles !{text} columns.

```
6382 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
6383 \LWR@traceinfo{bang column}%
6384 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token into \LWR@colparameter, expanding once:

```
6385 \LWR@traceinfo{about to read the next token:}%
6386 \expandarg%
6387 \StrChar{\LWR@origcolspec}%
6388     {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]
6389 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```
6390 \LWR@traceinfo{have now read the next token}%
6391 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
6392 {% left edge of the table:
6393     \LWR@traceinfo{at the left edge}%
6394     \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
6395 }%
6396 {% not at the left edge:
6397     \LWR@traceinfo{not at the left edge}%
6398     \LWR@setexparray{LWR@colbangspec}%
6399         {\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
6400     \LWR@traceinfo{bang \arabic{LWR@tabletotalcols}: \LWR@colparameter!}%
6401 }%
6402 \let\LWR@colparameter\relax%
6403 \booltrue{LWR@validtablecol}%
6404 }
```

\LWR@parsebeforecolumn Handles >{text} columns.

```
6405 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
6406 \addtocounter{LWR@tablecolspecindex}{1}%
```

Read the next token, expanding once into \LWR@colparameter:

```
6407 \expandarg%
6408 \StrChar{\LWR@origcolspec}%
6409     {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
6410 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@colparameter:

```

6411 \LWR@settexpparray{LWR@colbeforespec}%
6412     {\arabic{LWR@tabletotalcolsnext}}{\LWR@colparameter}%
6413 \let\LWR@colparameter\relax%
6414 \booltrue{LWR@validtablecol}%
6415 }

```

`\LWR@parseaftercolumn` Handles <{text} columns.

```

6416 \newcommand*{\LWR@parseaftercolumn}{%

```

Move to the next token after the '<':

```

6417 \addtocounter{LWR@tablecolspecindex}{1}%

```

Read the next token, expanding once into `\LWR@colparameter`:

```

6418 \expandarg%
6419 \StrChar{\LWR@origcolspec}%
6420     {\arabic{LWR@tablecolspecindex}}[\LWR@colparameter]%
6421 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@colparameter`:

```

6422 \LWR@settexpparray{LWR@colafterspec}%
6423     {\arabic{LWR@tabletotalcols}}{\LWR@colparameter}%
6424 \let\LWR@colparameter\relax%
6425 \booltrue{LWR@validtablecol}%
6426 }

```

`\LWR@parsebarcolumn` Handles vertical rules.

```

6427 \newcommand*{\LWR@parsebarcolumn}{%
6428 \LWR@traceinfo{LWR@parsebarcolumn}%

```

Remember the bar at this position:

```

6429 \ifnumcomp{\value{LWR@tabletotalcols}}{=}{0}%
6430 {% left edge of the table:
6431     \edef\LWR@tempone{\LWR@gettexpparray{LWR@colbarspec}{leftedge}}%
6432     \ifdefstring{\LWR@tempone}{tvertbarl}%
6433     {\LWR@settexpparray{LWR@colbarspec}{leftedge}{tvertbarldouble}}%
6434     {\LWR@settexpparray{LWR@colbarspec}{leftedge}{tvertbarl}}%
6435 }%
6436 {% not at the left edge:
6437     \edef\LWR@tempone{%
6438         \LWR@gettexpparray{LWR@colbarspec}{\arabic{LWR@tabletotalcols}}%

```

```

6439 }%
6440 \ifdefstring{\LWR@tempone}{tvertbarr}%
6441 {%
6442     \LWR@setexparray{\LWR@colbarspec}%
6443     {\arabic{\LWR@tabletotalcols}}{tvertbarrdouble}%
6444 }%
6445 {%
6446     \LWR@setexparray{\LWR@colbarspec}%
6447     {\arabic{\LWR@tabletotalcols}}{tvertbarr}%
6448 }%
6449 }%
6450 \booltrue{\LWR@validtablecol}%
6451 }

```

`\LWR@parsecoloncolumn` Handles vertical rules.

```

6452 \newcommand*{\LWR@parsecoloncolumn}{%
6453 \LWR@traceinfo{\LWR@parsecoloncolumn}%

```

Remember the bar at this position:

```

6454 \ifnumcomp{\value{\LWR@tabletotalcols}}{=}{0}%
6455 {% left edge of the table:
6456     \edef\LWR@tempone{\LWR@getexparray{\LWR@colbarspec}{leftedge}}%
6457     \ifdefstring{\LWR@tempone}{tvertbarldash}%
6458     {\LWR@setexparray{\LWR@colbarspec}{leftedge}{tvertbarldoubledash}}%
6459     {\LWR@setexparray{\LWR@colbarspec}{leftedge}{tvertbarldash}}%
6460 }%
6461 {% not at the left edge:
6462     \edef\LWR@tempone{%
6463         \LWR@getexparray{\LWR@colbarspec}{\arabic{\LWR@tabletotalcols}}%
6464     }%
6465     \ifdefstring{\LWR@tempone}{tvertbarrdash}%
6466     {\LWR@setexparray{\LWR@colbarspec}%
6467         {\arabic{\LWR@tabletotalcols}}{tvertbarrdoubledash}}%
6468     {\LWR@setexparray{\LWR@colbarspec}%
6469         {\arabic{\LWR@tabletotalcols}}{tvertbarrdash}}%
6470 }%
6471 \booltrue{\LWR@validtablecol}%
6472 }

```

`\LWR@parsesemicoloncolumn` Handles vertical rules.

```

6473 \newcommand*{\LWR@parsesemicoloncolumn}{%

```

Treat ; as a : column:

```

6474 \LWR@parsecoloncolumn%

```

Skip the following width token:

```
6475 \addtocounter{LWR@tablecolspecindex}{1}%
6476 }
```

68.9 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.

```
6477 \newcommand*{\LWR@parsenormalcolumn}[1]{%
6478 \appto\LWR@tablecolspec{#1}%
6479 \addtocounter{LWR@tablettotalcols}{1}%
6480 \addtocounter{LWR@tablettotalcolsnext}{1}%
6481 \LWR@traceinfo{normal column \arabic{LWR@tablettotalcols}: #1}%
6482 \LWR@setexparray{LWR@colatspec}{\arabic{LWR@tablettotalcolsnext}}{}%
6483 \LWR@setexparray{LWR@colbangspec}{\arabic{LWR@tablettotalcolsnext}}{}%
6484 \LWR@setexparray{LWR@colbeforespec}{\arabic{LWR@tablettotalcolsnext}}{}%
6485 \LWR@setexparray{LWR@colafterspec}{\arabic{LWR@tablettotalcolsnext}}{}%
6486 \LWR@setexparray{LWR@colbarspec}{\arabic{LWR@tablettotalcolsnext}}{}%
6487 \booltrue{LWR@validtablecol}%
6488 }
```

68.10 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\{ \langle thiscolumn \rangle \}$ The width will be ignored.

```
6489 \newcommand*{\LWR@parsepcolumn}[1]{%
```

Converts to the given column type:

```
6490 \LWR@parsenormalcolumn{#1}%
```

Skips the following width token:

```
6491 \addtocounter{LWR@tablecolspecindex}{1}%
6492 }
```

68.11 Parsing ‘w’ columns

`\LWR@parsewcolumn` The width will be ignored.

```
6493 \newcommand*{\LWR@parsewcolumn}{%
6494 \LWR@columnspeclookahead{1}%
6495 \expandafter\LWR@parsenormalcolumn\expandafter{\LWR@strresulttwo}%
```

Skips the following width and alignment tokens:

```
6496 \addtocounter{LWR@tablecolspecindex}{2}%
6497 }
```

68.12 Parsing ‘D’ columns

From the `dcolum` package.

`\LWR@parseDcolumn` `{\thiscolumn}` The three parameters will be ignored.

```
6498 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
6499 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
6500 \addtocounter{LWR@tablecolspecindex}{3}%
6501 }
```

68.13 Parsing the column specifications



HTML css cannot exactly match the \LaTeX concept of a baseline for a table row. Table 8 shows the \LaTeX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 9 for details.

Table 9 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

Table 8: Tabular baseline

l	p	m	b	r
---	---	---	---	---

			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

Table 9: Tabular HTML column conversions

l, r, c: Converted to table cells without paragraph tags.

Uses css `vertical-align:middle` so that top or bottom-aligned cells may go above or below this cell.

p: Converted to table cells with paragraph tags. Ref: Table 8, \LaTeX places the top line of a parbox aligned with the rest of the text line, so css `vertical-align:bottom` is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the css may be desired, changing `td.tdp`, `td.tdP`, `td.tdprule`, and `td.tdPrule` to `vertical-align: middle`. Another possibility is to change L,R,C, and P to `vertical-align: top` and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.

m: With paragraph tags, css `vertical-align:middle`.

b: With paragraph tags, css `vertical-align:top` so that the bottom of the text is closest to the middle of the text line.

P, M, B: Horizontally-centered versions.

S: Converted to 'r'. Ignores optional argument. From the **siunitx** package.

D: Converted to 'c'. From the **dcolum** package.

@, !, >, <: One each, in that order.

|: Vertical rule.

Unknown: Converted to 'l'.

\newcolumn: Currently treated as unknown.

```
6502 \newbool{LWR@validtablecol}
```

Bool LWR@opttablecol True if found a table column optional argument.

```
6503 \newbool{LWR@opttablecol}
```

\LWR@parsetablecols {<colspecs>}

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.

```
6504 \newcommand*{\LWR@parsetablecols}[1]{%
```

```
6505 \LWR@traceinfo{\LWR@parsetablecols}%
```

Remember the original supplied column spec:

```
6506 \renewcommand*{\LWR@origcolspec}{#1}%
```

Remove spaces:

```
6507 \expandarg%
```

```
6508 \StrSubstitute{\LWR@origcolspec}{ }{ }[\LWR@origcolspec]%
```

Clear the parsed resulting column spec:

```
6509 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
6510 \setcounter{LWR@tabletotalcols}{0}%
```

```
6511 \setcounter{LWR@tabletotalcolsnext}{1}%
```

```
6512 \LWR@setexparray{LWR@colatspec}{leftedge}{}%
```

```
6513 \LWR@setexparray{LWR@colatspec}{1}{}%
```

```
6514 \LWR@setexparray{LWR@colatspec}{2}{}%
```

```
6515 \LWR@setexparray{LWR@colatspec}{3}{}%
```

```
6516 \LWR@setexparray{LWR@colbangspec}{leftedge}{}%
```

```
6517 \LWR@setexparray{LWR@colbangspec}{1}{}%
```

```
6518 \LWR@setexparray{LWR@colbangspec}{2}{}%
```

```
6519 \LWR@setexparray{LWR@colbangspec}{3}{}%
```

```
6520 \LWR@setexparray{LWR@colbeforespec}{1}{}%
```

```
6521 \LWR@setexparray{LWR@colbeforespec}{2}{}%
```

```
6522 \LWR@setexparray{LWR@colbeforespec}{3}{}%
```

```
6523 \LWR@setexparray{LWR@colafterspec}{1}{}%
```

```
6524 \LWR@setexparray{LWR@colafterspec}{2}{}%
```

```

6525 \LWR@setexparray{LWR@colafterspec}{3}{}%
6526 \LWR@setexparray{LWR@colbarspec}{leftedge}{}%
6527 \LWR@setexparray{LWR@colbarspec}{1}{}%
6528 \LWR@setexparray{LWR@colbarspec}{2}{}%
6529 \LWR@setexparray{LWR@colbarspec}{3}{}%

```

Starting at the first column specification:

```

6530 \setcounter{LWR@tablecolspecindex}{1}%

```

Place the colspecs string length into \LWR@strresult, and remember the number of characters in the column specification:

```

6531 \expandarg%
6532 \StrLen{LWR@origcolspec}[LWR@strresult]%
6533 \fullexpandarg%
6534 \LWR@traceinfo{original column spec length: LWR@strresult}%
6535 \setcounter{LWR@tablecolspecwidth}{LWR@strresult}%

```

Haven't seen any optional arguments so far

```

6536 \boolfalse{LWR@opttablecol}%

```

Scan through the column specifications:

```

6537 \whileboolexpr{%
6538     not test{%
6539         \ifnumcomp{value{LWR@tablecolspecindex}}{>}%
6540             {value{LWR@tablecolspecwidth}}%
6541     }%
6542 }%
6543 {%

```

Place the next single-character column type into \LWR@strresult:

```

6544 \expandarg%
6545 \StrChar{LWR@origcolspec}{\arabic{LWR@tablecolspecindex}}[LWR@strresult]%
6546 \LWR@traceinfo{position \arabic{LWR@tablecolspecindex}: LWR@strresult}%
6547 \fullexpandarg%

```

Not yet found a valid column type:

```

6548 \boolfalse{LWR@validtablecol}%

```

Skip over any optional arguments, such as **siunitx** S column:

```

6549 \IfStrEq{LWR@strresult}{[]}{\booltrue{LWR@opttablecol}}{%

```

Throw away anything found inside the optional argument:

```
6550 \ifbool{LWR@opttablecol}%
6551 {}% inside an optional argument
6552 {}% not an optional tabular argument
```

Not inside an optional argument, so consider the column type:

```
6553 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
6554 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
6555 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
6556 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
6557 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
6558 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
6559 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%

6560 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{c}}{}%
6561 \IfStrEq{\LWR@strresult}{s}{\LWR@parsenormalcolumn{c}}{}%

6562 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
6563 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
6564 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
6565 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
6566 \IfStrEq{\LWR@strresult}{|}{\LWR@parsebarcolumn}{}%
6567 \IfStrEq{\LWR@strresult}{:}{\LWR@parsecoloncolumn}{}%
6568 \IfStrEq{\LWR@strresult}{;}{\LWR@parsesemicoloncolumn}{}%

6569 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
6570 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
6571 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

6572 \IfStrEq{\LWR@strresult}{w}{\LWR@parsewcolumn}{}%
6573 \IfStrEq{\LWR@strresult}{W}{\LWR@parsewcolumn}{}%
```

From the **dcolumn** package:

```
6574 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{}%
```

From the **tabularx** package. X column has no parameter, but will be given paragraph tags.

```
6575 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumnntype{P}[1]{>\centering\arraybackslash}p{#1}}
```

```

6576 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
6577 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
6578 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%

```

If this column was an invalid column type, convert it to an l column:

```

6579 \ifbool{\LWR@validtablecol}{}{}%
6580     \LWR@traceinfo{invalid column type: \LWR@strresult}%
6581     \LWR@parsenormalcolumn{l}%
6582 }%
6583 }% not an optional column argument

```

If read the closing bracket, no longer inside the optional argument:

```

6584 \IfStrEq{\LWR@strresult}{]}{\boolfalse{\LWR@opttablecol}}{}%

```

Move to the next character:

```

6585 \addtocounter{\LWR@tablecolspecindex}{1}%
6586 }% whiledo
6587 \LWR@traceinfo{\LWR@parsetablecols: Final table column spec: !\LWR@tablecolspec!}%
6588 }%

```

68.14 colortbl and xparse tabular color support

These macros provide a minimal emulation of some **colortbl** macros which might appear between table cells. If **colortbl** is loaded, these macros will be replaced with functional versions.

For each of the HTML colors below, the text for the HTML color is set if requested, but the macro is empty if none has been set.

\rownum Reserve a counter register.

```

6589 \@ifundefined{\rownum}{\newcount\rownum}{}

```

\@rowcolors Emulated in case **xcolor** is not used.

```

6590 \newcommand*{\@rowcolors}{}

```

\@rowc@lors Emulated in case **xcolor** is not used.

```

6591 \newcommand*{\@rowc@lors}{}

```

`\LWR@xcolorrowHTMLcolor` Emulated **xcolor** row color.

6592 `\newcommand*{\LWR@xcolorrowHTMLcolor}{}`

`\LWR@columnHTMLcolor` HTMLstyle code for the column color.

6593 `\newcommand*{\LWR@columnHTMLcolor}{}`

`\LWR@rowHTMLcolor` HTMLstyle code for the row color.

6594 `\newcommand*{\LWR@rowHTMLcolor}{}`

`\LWR@cellHTMLcolor` HTMLstyle code for the cell color.

6595 `\newcommand*{\LWR@cellHTMLcolor}{}`

`\LWR@ruleHTMLcolor` HTMLstyle code for the cell color.

6596 `\newcommand*{\LWR@ruleHTMLcolor}{}`

`\rowcolor` [*model*] {*color*} [*left overhang*] [*right overhang*] Print version. The HTML version is in **lwarp-colortbl**. Used before starting a tabular data cell, thus `\LWR@getmynexttoken`.

6597 `\newcommand*{\rowcolor}{\LWR@getmynexttoken}%`

`\arrayrulecolor` [*model*] {*color*}

`\arrayrulecolornexttoken` [*model*] {*color*}

Print versions for use outside and inside a tabular:

6598 `\newcommand{\arrayrulecolor}[2][named]{}`

6599 `\newcommand{\arrayrulecolornexttoken}[2][named]{\LWR@getmynexttoken}`

`\doublerulesepcolor` [*model*] {*color*}

`\doublerulesepcolornexttoken` [*model*] {*color*}

Print versions for use inside and outside a tabular:

6600 `\newcommand{\doublerulesepcolor}[2][named]{}`

6601 `\newcommand{\doublerulesepcolornexttoken}[2][named]{\LWR@getmynexttoken}`

68.15 Starting a new row

`\LWR@maybenewtablerow` If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for hline or tbrule if necessary.

```
6602 \newcommand*{\LWR@maybenewtablerow}
6603 {%
6604 \ifbool{\LWR@startedrow}%
6605 {}% started the row
6606 {}% not started the row
```

Remember that now have started the row:

```
6607 \global\booltrue{\LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
6608 \global\booltrue{\LWR@intabularmetadata}%
6609 \ifbool{expr{%
6610     test{\ifnumcomp{\value{\LWR@hlines}}{>}{0}} or%
6611     test{\ifnumcomp{\value{\LWR@hdashedlines}}{>}{0}}}%
6612 }%
6613 {%
6614     \LWR@htmltag{tr class="hline" }%
6615     \LWR@orignewline%
6616 }%
6617 {% not doing hline
6618     \ifbool{\LWR@doingtbrule}%
6619     {%
6620         \ifvoid{\LWR@ruleHTMLcolor}{%
6621             \LWR@htmltag{tr class="tbrule"}%
6622         }{%
6623             \LWR@htmltag{%
6624                 tr class="tbrule" % space
6625                 style="border-top: 1px solid % space
6626                     \LWR@origpound\LWR@ruleHTMLcolor "%
6627             }%
6628         }%
6629         \LWR@orignewline%
6630     }%
6631     {\LWR@htmltag{tr}\LWR@orignewline}%
6632 }% end of not doing hline
6633 }% end of not started the row
6634 }
```

68.16 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.

```

6635 \newcommand*\LWR@printbartag}[1]{%
6636 \LWR@traceinfo{\LWR@printbartag !#1!}%
6637 \ifboolexpr{bool{\LWR@tabularmutemods} or bool{\LWR@emptyatbang}}%
6638 {}% muting or empty
6639 {% not muting
6640     \edef\LWR@tempone{\LWR@getexpparray{\LWR@colbarspec}{#1}}%
6641     \ifdefempty{\LWR@tempone}{\LWR@tempone}%
6642 }% not muting
6643 \LWR@traceinfo{\LWR@printbartag done}%
6644 }
```

68.17 Printing at or bang tags

`\LWR@printatbang` $\{\langle at -or- bang \rangle\} \{\langle index \rangle\}$

```

6645 \newcommand*\LWR@printatbang}[2]{%
```

Fetch the column at or bang spec:

```

6646 \edef\LWR@atbangspec{\LWR@getexpparray{\LWR@col#1spec}{#2}}%
6647 \LWR@traceinfo{atbang: #2 !\LWR@atbangspec!}%
```

Only generate if is not empty;

```

6648 \ifdefempty{\LWR@atbangspec}%
6649 {}%
6650 {% not empty
6651     \LWR@htmltag{%
6652         td class="td#1%
6653         \LWR@subaddcmidruletrim{}{}%
6654         \LWR@printbartag{#2}%
6655         "%
6656         \LWR@tdstartstyles%
6657         \LWR@addcmidrulewidth%
6658         \LWR@addcdashline%
6659         \LWR@addtabularrulecolors%
6660         \LWR@tdendstyles%
6661     }%
```

Create an empty cell if muting for the \bottomrule:

```

6662 \ifbool{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}{%
6663   {}%
6664   {\LWR@atbangspec}%
6665 %
6666 \LWR@htmltag{/td}\LWR@orignewline%
6667 \global\booltrue{LWR@tabularcelladded}%
6668 }% not empty
6669 }%

```

\LWR@addleftmostbartag

```

6670 \newcommand*{\LWR@addleftmostbartag}{%
6671 \ifnumcomp{\value{LWR@tablecolindex}}{=}{1}{%
6672   \LWR@printbartag{leftedge}%
6673 }{}%
6674 }

```

\LWR@tabularleftedge

```

6675 \newcommand*{\LWR@tabularleftedge}{%
6676 \ifnumcomp{\value{LWR@tablecolindex}}{=}{1}%
6677 {%
6678   \LWR@printatbang{at}{leftedge}%
6679   \LWR@printatbang{bang}{leftedge}%
6680 }% left edge
6681 {}% not left edge
6682 }

```

68.18 Data opening tag

\LWR@thiscolspec Temporary storage.

```

6683 \newcommand*{\LWR@thiscolspec}{}

```

\LWR@tabledatasinglecolumnntag Print a table data opening tag with style for alignment and color.

```

6684 \newcommand*{\LWR@tabledatasinglecolumnntag}{%
6685 {%
6686 \LWR@traceinfo{LWR@tabledatasinglecolumnntag}%
6687 \LWR@maybenewtablerow%

```

Don't start a new paragraph tag if have already started one:

```
6688 \ifbool{LWR@intabularmetadata}%
6689 {%
```

If have found the end of tabular command, do not create the next data cell:

```
6690 \ifbool{LWR@exitingtabular}{}%
6691 {% not exiting tabular
```

Print the @ and ! contents before first column:

```
6692 \LWR@tabularleftedge%
```

Fetch the current column's alignment character into \LWR@strresult:

```
6693 \StrChar{\LWR@tablecolspec}%
6694 {\arabic{\LWR@tablecolindex}}[\LWR@strresult]%
```

print the start of a new table data cell:

```
6695 \LWR@traceinfo{\LWR@tabledatasinglecolumn: about to print td tag}%
6696 \LWR@htmltag{td class="td%
```

append this column's spec:

```
6697 \LWR@strresult%
```

If this column has a cmidrule, add “rule” to the end of the HTML class tag. Also add vertical bar tags.

```
6698 \LWR@addcmidruletrim%
6699 \LWR@addleftmostbartag%
6700 \LWR@printbartag{\arabic{\LWR@tablecolindex}}%
6701 "%
```

Add styles for rules, alignment:

```
6702 \LWR@tdstartstyles%
6703 \LWR@addcmidrulewidth%
6704 \LWR@addcdashline%
6705 \StrChar{\LWR@tablecolspec}%
6706 {\arabic{\LWR@tablecolindex}}[\LWR@thiscolspec]%
6707 \LWR@addformatwpaignment{\LWR@thiscolspec}%
```

Add styles for cell and rule colors:

```

6708      \LWR@addtabulararrowcolor%
6709      \LWR@addtabularrulecolors%

6710      \LWR@tdendstyles%
6711      }%
6712      \LWR@traceinfo{LWR@tabledatasinglecolumn tag: done printing td tag}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

6713      \ifboolexpr{%
6714          test{ \ifdefstring{LWR@strresult}{p} } or
6715          test{ \ifdefstring{LWR@strresult}{m} } or
6716          test{ \ifdefstring{LWR@strresult}{b} } or
6717          test{ \ifdefstring{LWR@strresult}{P} } or
6718          test{ \ifdefstring{LWR@strresult}{M} } or
6719          test{ \ifdefstring{LWR@strresult}{B} } or
6720          test{ \ifdefstring{LWR@strresult}{X} }
6721      }%
6722      {% allow pars
6723          \LWR@traceinfo{LWR@tabledatasinglecolumn tag: about to LWR@startpars}%
6724          \global\booltrue{LWR@tableparcell}%
6725          \LWR@startpars%
6726          \LWR@traceinfo{LWR@tabledatasinglecolumn tag: done with LWR@startpars}%
6727      }% allow pars
6728      }% no pars

```

Print the > contents unless muted for the \bottomrule:

```

6729      \ifboolexpr{bool{LWR@tabularmutemods} or bool{LWR@emptyatbang}}%
6730      {%
6731      {%
6732          \LWR@getexparray{LWR@colbefore spec}{\arabic{LWR@tablecolindex}}%
6733      }%
6734      \global\boolfalse{LWR@intabularmetadata}%
6735      }% not exiting tabular
6736 }{% in tabular metadata
6737 \LWR@traceinfo{LWR@tabledatasinglecolumn tag: done}%
6738 }%

```

68.19 Midrules

LWR@midrules LWR@midrules is a data array (section 39) of columns each containing a non-zero width if a midrule should be created for this column.

LWR@trimlrules LWR@trimlrules is a data array (section 39) of columns containing 1 if a midrule should be left trimmed for each column.

LWR@trimrrules LWR@trimrrules is a data array (section 39) of columns containing r if a midrule should be right trimmed for each column.

LWR@cdashlines LWR@cdashlines is a data array (section 39) of columns each containing a Y if an **arydshln** package "cdashed line" should be created for this column.

Ctr LWR@midrulecounter Indexes across the LWR@midrules and LWR@trim<l/r>rules data arrays.

```
6739 \newcounter{LWR@midrulecounter}
```

Len \LWR@heavyrulewidth The default width of the rule.

```
6740 \newlength{\LWR@heavyrulewidth}
6741 \setlength{\LWR@heavyrulewidth}{.08em}
```

Len \LWR@lightrulewidth The default width of the rule.

```
6742 \newlength{\LWR@lightrulewidth}
6743 \setlength{\LWR@lightrulewidth}{.05em}
```

Len \LWR@cmidrulewidth The default width of the rule.

```
6744 \newlength{\LWR@cmidrulewidth}
6745 \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.

```
6746 \newlength{\LWR@thiscmidrulewidth}
6747 \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.

```
6748 \newcommand*{\LWR@clearmidrules}
6749 {%
6750 \setcounter{LWR@midrulecounter}{1}%
6751 \whileboolexpr{%
6752   not test{%
6753     \ifnumcomp{\value{LWR@midrulecounter}}{>}%
6754       {\value{LWR@tablecolspecwidth}}%
6755   }%
6756 }%
6757 {%
```

```

6758 \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{Opt}%
6759 \setlength{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
6760 \LWR@setexparray{LWR@trimlrules}{\arabic{LWR@midrulecounter}}{}%
6761 \LWR@setexparray{LWR@trimrrules}{\arabic{LWR@midrulecounter}}{}%
6762 \LWR@setexparray{LWR@cdashlines}{\arabic{LWR@midrulecounter}}{N}%
6763 \addtocounter{LWR@midrulecounter}{1}%
6764 }%
6765 }

```

`\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.

```

6766 \newcommand*{\LWR@subcmidrule}[4]{%
6767 \setcounter{LWR@midrulecounter}{#3}%
6768 \whileboolexpr{%
6769     not test {%
6770         \ifnumcomp{\value{LWR@midrulecounter}}{>}{#4}%
6771     }%
6772 }%
6773 {%
6774     \LWR@setexparray{LWR@midrules}{\arabic{LWR@midrulecounter}}{#1}%
6775     \addtocounter{LWR@midrulecounter}{1}%
6776 }% \whiledo
6777 \IfSubStr{#2}{l}{\LWR@setexparray{LWR@trimlrules}{#3}{l}}{}%
6778 \IfSubStr{#2}{r}{\LWR@setexparray{LWR@trimrrules}{#4}{r}}{}%
6779 \booltrue{LWR@doingcmidrule}%
6780 }

```

`\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.

```

6781 \NewDocumentCommand{\LWR@docmidrule}
6782 {0{\LWR@cmidrulewidth} D{}} >\SplitArgument{1}{-}m
6783 {\LWR@subcmidrule{#1}{#2}#3}

```

`\LWR@subcdashline` $\{ \langle leftcolumn \rangle \} \{ \langle rightcolumn \rangle \}$

Marks LWR@cdashlines data array elements to be Y from left to right columns.

LWR@doingcmidrule is set to force an empty row at the end of the tabular to create the rule.

```

6784 \newcommand*{\LWR@subcdashline}[2]{%
6785 \setcounter{LWR@midrulecounter}{#1}%
6786 \whileboolexpr{%
6787     not test {%
6788         \ifnumcomp{\value{LWR@midrulecounter}}{>}{#2}%
6789     }%
6790 }%
6791 {%
6792     \LWR@setexparray{LWR@cdashlines}{\arabic{LWR@midrulecounter}}{Y}%
6793     \addtocounter{LWR@midrulecounter}{1}%
6794 }% \whiledo
6795 \booltrue{LWR@doingcmidrule}%
6796 }
```

`\LWR@docdashline` `{\leftcolumn-rightcolumn}`

Marks LWR@cdashlines data array elements to be Y from left to right columns.

```

6797 \NewDocumentCommand{\LWR@docdashline}
6798     {>\SplitArgument{1}{-}}m{%
6799     {%
6800         \LWR@subcdashline#1%
6801     }
```

Used to compute margins, tabular trims, column offsets:

```

6802 \newlength{\LWR@templengthone}
6803 \newlength{\LWR@templengthtwo}
6804 \newlength{\LWR@templengththree}
6805 \newcounter{LWR@tempcountone}
```

Used to add a style to a table data cell:

```

6806 \newboolean{LWR@tdhavecellstyle}
```

`\LWR@tdstartstyles` Begins possibly adding a table data cell style.

```

6807 \newcommand*{\LWR@tdstartstyles}{\global\boolfalse{LWR@tdhavecellstyle}}
```

`\LWR@tdaddstyle` Starts adding a table data cell style.

```

6808 \newcommand*{\LWR@tdaddstyle}{%
6809 \ifbool{LWR@tdhavecellstyle}%
```

```

6810 {; }%
6811 { style="%}%
6812 \booltrue{LWR@tdhavecellstyle}%
6813 }

```

`\LWR@tdendstyles` Finishes possibly adding a table data cell style. Prints the closing quote.

```

6814 \newcommand*{\LWR@tdendstyles}{%
6815 \ifbool{LWR@tdhavecellstyle}{%
6816     "%
6817     \global\boolfalse{LWR@tdhavecellstyle}%
6818 }{}%
6819 }

```

`\LWR@subaddcmidruletrim` `{<lefttrim>} {<righttrim>}` Adds a `\cmidrule` with optional trim.

```

6820 \newcommand*{\LWR@subaddcmidruletrim}[2]{%
6821 \setlength{\LWR@templengthone}{%
6822     \LWR@getexparray{LWR@midrules}{\arabic{LWR@tablecolindex}}}%
6823 }%
6824 \ifdimcomp{\LWR@templengthone}{>}{0pt}%
6825 {%

```

Print the class with left and right trim letters appended:

```

6826     \LWR@origtilde tdrule#1#2%

```

Remember the width of the rule:

```

6827     \setlength{\LWR@thiscmidrulewidth}{\LWR@templengthone}%
6828 }%
6829 {%
6830     \setlength{\LWR@thiscmidrulewidth}{0pt}%
6831 }%
6832 }

```

`\LWR@addcmidruletrim` Adds left or right trim to a `\cmidrule`.

```

6833 \newcommand*{\LWR@addcmidruletrim}{%
6834 \LWR@subaddcmidruletrim%
6835 {\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
6836 {\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@tablecolindex}}}%
6837 }

```

`\LWR@addrulewidth` `{<thiswidth>} {<defaultwidth>}`

If not default width, add a custom style with width and color depending on `thiswidth`.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6838 \newcommand{\LWR@addrulewidth}[2]{%
```

Only add a custom width if `thiswidth` is different than the `defaultwidth`, or if a color is being used:

```
6839 \ifboolexpr{%
6840   test{\ifdimcomp{#1}={}\{Opt}\} or
6841   (
6842     ( test{\ifdimcomp{#1}={}\{#2}\} and not bool{FormatWP} )
6843     and ( test {\ifdefvoid{\LWR@ruleHTMLcolor}\} )
6844   )
6845 }%
6846 {}% default width and color
6847 {}% custom width and/or color
```

Ensure that the width is wide enough to display in the browser:

```
6848   \LWR@forceminwidth{#1}%
```

Begin adding another style:

```
6849   \LWR@tdaddstyle%
```

The style itself:

```
6850   border-top:\LWR@printlength{\LWR@atleastonept} solid %
```

If default gray, the darkness of the color depends on the thickness of the rule:

```
6851   \ifdefvoid{\LWR@ruleHTMLcolor}{%
6852     \ifdimcomp{#1}<{\LWR@lightrulewidth}%
6853     {\LWR@origpound{ }AOAOAO}%
6854     {% lightrule or heavier
6855       \ifdimcomp{#1}<{\LWR@heavyrulewidth}%
6856       {\LWR@origpound{ }808080}%
6857       {black}%
6858     }% lightrule or heavier
6859   }{%
6860     \LWR@origpound\LWR@ruleHTMLcolor%
6861   }
6862 }% custom width and/or color
6863 }
```

`\LWR@addcmidrulewidth` Adds a style for the rule width.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6864 \newcommand{\LWR@addcmidrulewidth}{%
6865 \LWR@addrulewidth{\LWR@thiscmidrulewidth}{\LWR@cmidrulewidth}%
6866 }
```

`\LWR@addcdashline` Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6867 \newcommand{\LWR@addcdashline}{%
6868 \edef\LWR@tempone{%
6869   \LWR@getexparray{\LWR@cdashlines}{\arabic{\LWR@tablecolindex}}%
6870 }%
6871 \ifdefstring{\LWR@tempone}{Y}{%
6872   \LWR@tdaddstyle%
6873     border-top: 1pt dashed %
6874   \ifdefvoid{\LWR@ruleHTMLcolor}%
6875     {black}%
6876   {\LWR@origpound\LWR@ruleHTMLcolor}%
6877 }{}%
6878 }
```

`\LWR@WPcell` `{<text-align>}{<vertical-align>}`

```
6879 \newcommand*{\LWR@WPcell}[2]{%
6880 \LWR@tdaddstyle%
6881 \LWR@print@mbbox{text-align:#1}; \LWR@print@mbbox{vertical-align:#2}%
6882 }
```

`\LWR@addformatwpalignment` If `FormatWP`, adds a style for the alignment.

Must be placed between `\LWR@tdstartstyles` and `\LWR@tdendstyles`.

```
6883 \newcommand*{\LWR@addformatwpalignment}[1]{%
6884 \ifbool{FormatWP}{%
6885 \IfSubStr{#1}{l}{\LWR@WPcell{left}{middle}}{}%
6886 \IfSubStr{#1}{c}{\LWR@WPcell{center}{middle}}{}%
6887 \IfSubStr{#1}{r}{\LWR@WPcell{right}{middle}}{}%
6888 \IfSubStr{#1}{p}{\LWR@WPcell{left}{bottom}}{}%
6889 \IfSubStr{#1}{m}{\LWR@WPcell{left}{middle}}{}%
6890 \IfSubStr{#1}{b}{\LWR@WPcell{left}{top}}{}%
6891 \IfSubStr{#1}{P}{\LWR@WPcell{center}{bottom}}{}%
6892 \IfSubStr{#1}{M}{\LWR@WPcell{center}{middle}}{}%
6893 \IfSubStr{#1}{B}{\LWR@WPcell{center}{top}}{}%
6894 }{}%
6895 }
```

68.20 Cell colors

`\LWR@addtabularrowcolor` Adds a cell's row color style, if needed.

No color is added for the final row of empty cells which finishes each tabular.

```

6896 \newcommand*{\LWR@addtabularrowcolor}{%
6897 \ifbool{\LWR@tabularmutemods}{}%
6898   \ifdefvoid{\LWR@rowHTMLcolor}{%
6899     \ifdefvoid{\LWR@xcolorrowHTMLcolor}{}%
6900     {% xcolor row color
6901       \LWR@tdaddstyle%
6902       background:\LWR@origpound\LWR@xcolorrowHTMLcolor%
6903     }%
6904   }%
6905   {% explicit row color
6906     \LWR@tdaddstyle%
6907     background:\LWR@origpound\LWR@rowHTMLcolor%
6908   }%
6909 }%
6910 }
```

`\LWR@addtabularhrulecolor` Adds a cell's horizontal rule color style, if needed.

```

6911 \newcommand*{\LWR@addtabularhrulecolor}{%
```

If either form of horizontal rule is requested:

```

6912 \ifboolexpr{%
6913   test{\ifnumcomp{\value{\LWR@hlines}}{>}{0}} or%
6914   test{\ifnumcomp{\value{\LWR@hdashedlines}}{>}{0}} or%
6915   bool{\LWR@doingtbrule}%
6916 }{%
```

If there is a no custom color:

```

6917   \ifdefvoid{\LWR@ruleHTMLcolor}%
6918   {%
6919     \ifnumcomp{\value{\LWR@hlines}}{>}{1}%
6920     {%
6921       \LWR@tdaddstyle%
6922       border-top: 4px double%
6923     }{% else
6924       \ifnumcomp{\value{\LWR@hdashedlines}}{>}{1}%
6925       {%
6926         \LWR@tdaddstyle%
6927         border-top: 2px dashed%
6928       }{% else
```

```

6929      \ifnumcomp{\value{LWR@hdashedlines}}{=}{1}%
6930      {%
6931          \LWR@tdaddstyle%
6932          border-top: 1px dashed%
6933      }{}%

```

If no color and not doubled or dashed, then add nothing, since a simpler rule is the default.

```

6934      }%

```

If there is a custom color:

```

6935      {%
6936      \ifnumcomp{\value{LWR@hlines}}{>}{1}%
6937      {%
6938          \LWR@tdaddstyle%
6939          border-top: 4px double \LWR@origpound\LWR@ruleHTMLcolor%
6940      }{% else
6941      \ifnumcomp{\value{LWR@hdashedlines}}{>}{1}%
6942      {%
6943          \LWR@tdaddstyle%
6944          border-top: 2px dashed \LWR@origpound\LWR@ruleHTMLcolor%
6945      }{% else
6946      \ifnumcomp{\value{LWR@hdashedlines}}{=}{1}%
6947      {%
6948          \LWR@tdaddstyle%
6949          border-top: 1px dashed \LWR@origpound\LWR@ruleHTMLcolor%
6950      }{% else
6951          \LWR@tdaddstyle%
6952          border-top: 1px solid \LWR@origpound\LWR@ruleHTMLcolor%
6953      }{}%
6954      }%
6955      }{}%
6956      }

```

`\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.

```

6957 \newcommand*{\LWR@addtabularrulecolors}{%

```

Custom horizontal rule color:

```

6958 \LWR@addtabularhrulecolor%

```

No vertical rules if finishing the tabular with a row of empty cells:

```
6959 \ifbool{LWR@tabularmutemods}{}{%
```

If at the leftmost cell, possibly add a leftmost vertical rule:

```
6960 \ifnumequal{\value{LWR@tablecolindex}}{1}{%
```

Fetch the left edge's vertical bar specification:

```
6961 \edef\LWR@tempone{\LWR@getexparray{LWR@colbarspec}{leftedge}}%
```

Add a custom style if a vertical bar was requested:

```
6962 \ifdefstring{\LWR@tempone}{tvertbarl}{%
6963     \LWR@tdaddstyle%
6964     border-left: 1px solid % space
6965     \LWR@origpound\LWR@vertruleHTMLcolor%
6966 }{}%
6967 \ifdefstring{\LWR@tempone}{tvertbarldouble}{%
6968     \LWR@tdaddstyle%
6969     border-left: 4px double % space
6970     \LWR@origpound\LWR@vertruleHTMLcolor%
6971 }{}%
6972 \ifdefstring{\LWR@tempone}{tvertbarldash}{%
6973     \LWR@tdaddstyle%
6974     border-left: 1px dashed % space
6975     \LWR@origpound\LWR@vertruleHTMLcolor%
6976 }{}%
6977 \ifdefstring{\LWR@tempone}{tvertbarldoubledash}{%
6978     \LWR@tdaddstyle%
6979     border-left: 2px dashed % space
6980     \LWR@origpound\LWR@vertruleHTMLcolor%
6981 }{}%
6982 }{}%
```

Possibly add a right vertical rule for this cell:

```
6983 \edef\LWR@tempone{%
6984     \LWR@getexparray{LWR@colbarspec}{\arabic{LWR@tablecolindex}}%
6985 }%
6986 \ifdefstring{\LWR@tempone}{tvertbarr}{%
```

Add a custom style if a vertical bar was requested:

```
6987     \LWR@tdaddstyle%
6988     border-right: 1px solid \LWR@origpound\LWR@vertruleHTMLcolor%
6989 }{}%
6990 \ifdefstring{\LWR@tempone}{tvertbarrdouble}{%
6991     \LWR@tdaddstyle%
```

```

6992         border-right: 4px double \LWR@origpound\LWR@vertruleHTMLcolor%
6993     }{}%
6994     \ifdefstring{\LWR@tempone}{tvertbarrdash}{%
6995         \LWR@tdaddstyle%
6996         border-right: 1px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
6997     }{}%
6998     \ifdefstring{\LWR@tempone}{tvertbarrdoubledash}{%
6999         \LWR@tdaddstyle%
7000         border-right: 2px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
7001     }{}%
7002 }%
7003 }

```

Ctr LWR@cellcolordepth Counts how many cell color <div>s were added to the current tabular data cell.

```
7004 \newcounter{LWR@cellcolordepth}
```

\LWR@subadddtabularcellcolor {<HTML color>}

```

7005 \newcommand*{\LWR@subadddtabularcellcolor}[1]{%
7006 \LWR@htmltag{div class="cellcolor" style="%
7007     background:\LWR@origpound{ }#1 %
7008 " }%
7009 \addtocounter{LWR@cellcolordepth}{1}%
7010 }

```

\LWR@addtabularcellcolor Adds a cell color style, if needed.

```

7011 \newcommand*{\LWR@addtabularcellcolor}{%
7012 \ifdefvoid{\LWR@cellHTMLcolor}%
7013 {%
7014     \ifdefvoid{\LWR@rowHTMLcolor}%
7015     {%
7016         \ifdefvoid{\LWR@xcolorrowHTMLcolor}%
7017         {%
7018             \ifdefvoid{\LWR@columnHTMLcolor}%
7019             {}%
7020             {\LWR@subadddtabularcellcolor{\LWR@columnHTMLcolor}}%
7021         }%
7022         {\LWR@subadddtabularcellcolor{\LWR@xcolorrowHTMLcolor}}%
7023     }%
7024     {\LWR@subadddtabularcellcolor{\LWR@rowHTMLcolor}}%
7025 }%
7026 {\LWR@subadddtabularcellcolor{\LWR@cellHTMLcolor}}%
7027 }

```

68.21 Multicolumns

68.21.1 Parsing multicolumns

7028 \newcounter{LWR@tablemulticolwidth}

Indexes into the multicolumn specification:

7029 \newcounter{LWR@tablemulticolspos}

Remembers multicolumn vertical rules if found in the column spec.

7030 \newcounter{LWR@mcolvertbarsl}

7031 \newcounter{LWR@mcolvertbarsr}

7032 \newcounter{LWR@mcolvertbarsldash}

7033 \newcounter{LWR@mcolvertbarsrdash}

7034 \newboolean{LWR@mcolvertbaronleft}%

\LWR@printmccoltype $\{\langle colspec \rangle\}$ Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag's class.

7035 \newcommand*{\LWR@printmccoltype}[1]{%

7036 \LWR@traceinfo{lw@printmccoltype -#1-}%

Get one token of the column spec:

7037 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%

Add to the HTML tag depending on which column type is found:

7038 \IfStrEq{\LWR@strresult}{l}{l}{}%

7039 \IfStrEq{\LWR@strresult}{c}{c}{}%

7040 \IfStrEq{\LWR@strresult}{r}{r}{}%

7041 \IfStrEq{\LWR@strresult}{p}{p}{}%

7042 \IfStrEq{\LWR@strresult}{m}{m}{}%

7043 \IfStrEq{\LWR@strresult}{b}{b}{}%

7044 \IfStrEq{\LWR@strresult}{P}{P}{}%

7045 \IfStrEq{\LWR@strresult}{M}{M}{}%

7046 \IfStrEq{\LWR@strresult}{B}{B}{}%

7047 \IfStrEq{\LWR@strresult}{w}{w}{}%

7048 \IfStrEq{\LWR@strresult}{W}{W}{}%

7049 \IfStrEq{\LWR@strresult}{S}{c}{}%

7050 \IfStrEq{\LWR@strresult}{s}{c}{}%

```

7051 \IfStrEq{\LWR@strresult}{X}{p}{}%

7052 \IfStrEq{\LWR@strresult}{|}%
7053 {%
7054     \ifbool{\LWR@mcolvertbaronleft}%
7055         {\addtocounter{\LWR@mcolvertbarsl}{1}}% left edge
7056         {\addtocounter{\LWR@mcolvertbarsr}{1}}% not left edge
7057 }%
7058 {%
7059     \IfStrEq{\LWR@strresult}{:}%
7060     {%
7061         \ifbool{\LWR@mcolvertbaronleft}%
7062             {\addtocounter{\LWR@mcolvertbarsldash}{1}}% left edge
7063             {\addtocounter{\LWR@mcolvertbarsrdash}{1}}% not left edge
7064     }%
7065     {%
7066         \IfStrEq{\LWR@strresult}{;}%
7067         {%
7068             \ifbool{\LWR@mcolvertbaronleft}%
7069                 {\addtocounter{\LWR@mcolvertbarsldash}{1}}% left edge
7070                 {\addtocounter{\LWR@mcolvertbarsrdash}{1}}% not left edge
7071             }%
7072             {\setboolean{\LWR@mcolvertbaronleft}{false}}%
7073         }%
7074     }%
7075 \LWR@traceinfo{lw@printmccoltype done}%
7076 }

```

`\LWR@mcolpartext` $\{\langle num\ parameters\rangle\}$ Print the data with paragraph tags, advance to bypass the given number of parameters.

```

7077 \newcommand*{\LWR@mcolpartext}[1]{%
7078 \LWR@startpars%
7079 \LWR@mcoltext%
7080 \addtocounter{\LWR@tablemulticolspos}{#1}%
7081 \LWR@stoppars%
7082 }

```

`\LWR@mcolother` $\{\langle colspec\rangle\}$ For @, !, >, <, print the next token without paragraph tags:

```

7083 \newcommand*{\LWR@mcolother}[1]{%
7084 \addtocounter{\LWR@tablemulticolspos}{1}%
7085 \StrChar{#1}{\arabic{\LWR@tablemulticolspos}}[\LWR@strresult]%
7086 \LWR@strresult%

```

A valid column data type was found:

```
7087 \booltrue{LWR@validtablecol}%
7088 }
```

\LWR@multicolskip Nothing to print for this column type.

```
7089 \newcommand*{\LWR@multicolskip}{%
```

A valid column data type was found:

```
7090 \booltrue{LWR@validtablecol}%
7091 }
```

\LWR@printmccoldata $\{ \langle colspec \rangle \}$ Print the data for any valid column type found.

```
7092 \newcommand*{\LWR@printmccoldata}[1]{%
7093 \LWR@traceinfo{lwarp@printmccoldata -#1}%
```

Not yet found a valid column type:

```
7094 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
7095 \StrChar{#1}{\arabic{LWR@tablemulticolspos}}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
7096 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
7097 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
7098 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
7099 \IfStrEq{\LWR@strresult}{D}{}%
7100 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
7101 \LWR@multicoltext%
7102 }{}}%
```

```
7103 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext{0}}{}}%
7104 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext{0}}{}}%
7105 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext{0}}{}}%
7106 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext{0}}{}}%
7107 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext{0}}{}}%
7108 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext{0}}{}}%
```

```
7109 \IfStrEq{\LWR@strresult}{w}{\LWR@multicolparttext{3}}{}}%
7110 \IfStrEq{\LWR@strresult}{W}{\LWR@multicolparttext{3}}{}}%
```

```

7111 \IfStrEq{\LWR@strresult}{S}{\LWR@multicoltext}{}%
7112 \IfStrEq{\LWR@strresult}{s}{\LWR@multicoltext}{}%

7113 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext{0}}{%
7114 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
7115 \IfStrEq{\LWR@strresult}{:}{\LWR@multicolskip}{}%
7116 \IfStrEq{\LWR@strresult}{;}{%
7117     \LWR@multicolskip%
7118     \addtocounter{LWR@tablemulticolspos}{1}% skip parameter
7119 }{%
7120 \IfStrEq{\LWR@strresult}{\detokenize@}{\LWR@multicolother{#1}}{%
7121 \IfStrEq{\LWR@strresult}{\detokenize!}{\LWR@multicolother{#1}}{%
7122 \IfStrEq{\LWR@strresult}{\detokenize>}{\LWR@multicolother{#1}}{%
7123 \IfStrEq{\LWR@strresult}{\detokenize<}{\LWR@multicolother{#1}}{%

```

If an invalid column type:

```

7124 \ifbool{LWR@validtablecol}{\LWR@multicoltext}%

```

Tracing:

```

7125 \LWR@traceinfo{lw@printmccoldata done}%
7126 }

```

`\parsemulticolumnalignment` $\{\langle 1: \textit{colspec} \rangle\} \{\langle 2: \textit{printresults} \rangle\}$

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.

```

7127 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
7128 \setcounter{LWR@tablemulticolspos}{1}%
7129 \StrLen{#1}[\LWR@strresult]%
7130 \setcounter{LWR@tablemulticolswidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

7131 \whilebool{expr}%
7132     not test {%
7133         \ifnumcomp{\value{LWR@tablemulticolspos}}{>}%
7134             {\value{LWR@tablemulticolswidth}}%
7135         }%
7136 }%
7137 {%

```

Execute the assigned print function for each token in the column spec:

7138 #2{#1}%

Move to the next token in the column spec:

```
7139 \addtocounter{LWR@tablemulticolspos}{1}%
7140 }%
7141 }
```

68.21.2 Multicolumn factored code

\LWR@addmulticolvertrulecolor

```
7142 \newcommand*{\LWR@addmulticolvertrulecolor}{%
```

No vertical rules if finishing the tabular with a row of empty cells:

```
7143 \ifbool{LWR@tabularmutemods}{}{%
```

Left side:

```
7144 \ifnumcomp{\value{LWR@mcolvertbarsl}}{=}{1}{%
7145 \LWR@tdaddstyle%
7146 border-left: 1px solid \LWR@origpound\LWR@vertruleHTMLcolor%
7147 }{}%
7148 \ifnumcomp{\value{LWR@mcolvertbarsl}}{>}{1}{%
7149 \LWR@tdaddstyle%
7150 border-left: 4px double \LWR@origpound\LWR@vertruleHTMLcolor%
7151 }{}%
7152 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{=}{1}{%
7153 \LWR@tdaddstyle%
7154 border-left: 1px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
7155 }{}%
7156 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{>}{1}{%
7157 \LWR@tdaddstyle%
7158 border-left: 2px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
7159 }{}%
```

Right side:

```
7160 \ifnumcomp{\value{LWR@mcolvertbarsr}}{=}{1}{%
7161 \LWR@tdaddstyle%
7162 border-right: 1px solid \LWR@origpound\LWR@vertruleHTMLcolor%
7163 }{}%
7164 \ifnumcomp{\value{LWR@mcolvertbarsr}}{>}{1}{%
7165 \LWR@tdaddstyle%
7166 border-right: 4px double \LWR@origpound\LWR@vertruleHTMLcolor%
7167 }{}%
```

```

7168 \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{=}{1}{%
7169 \LWR@tdaddstyle%
7170 border-right: 1px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
7171 }{}%
7172 \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{>}{1}{%
7173 \LWR@tdaddstyle%
7174 border-right: 2px dashed \LWR@origpound\LWR@vertruleHTMLcolor%
7175 }{}%
7176 }%
7177 }

```

```
7178 \newcommand{\LWR@multicoltext}{}

```

To find multicolumn right trim:

```
7179 \newcounter{LWR@lastmulticolumn}

```

```

\LWR@domulticolumn {[\langle 1: vpos\rangle] [\langle 2: #rows\rangle] {\langle 3: numLaTeXcols\rangle} {\langle 4: numHTMLcols\rangle} {\langle 5: colspec\rangle}
{\langle 6: text\rangle}

```

```

7180 \NewDocumentCommand{\LWR@domulticolumn}{o o m m m +m}{%
7181 \LWR@traceinfo{\LWR@domulticolumn -#1- -#2- -#4- -#5-}%

```

Remember the text to be inserted, and remember that a valid column type was found:

```

7182 \renewcommand{\LWR@multicoltext}{%
7183 #6%
7184 \booltrue{LWR@validtablecol}%
7185 }%

```

Compute the rightmost column to be included. This is used to create the right trim.

```

7186 \setcounter{LWR@lastmulticolumn}{\value{LWR@tablecolindex}}%
7187 \addtocounter{LWR@lastmulticolumn}{#3}%
7188 \addtocounter{LWR@lastmulticolumn}{-1}%

```

Row processing:

```
7189 \LWR@maybenewtablerow%

```

Begin the opening table data tag:

```
7190 \LWR@htmltag{td colspan="#4" %

```

```

7191 \IfValueT{#2}{ % rows?
7192 rowspan="#2" %

```

```

7193 \IfValueT{#1}{% vpos?
7194 \ifstrequal{#1}{b}{style="\LWR@print@mbx{vertical-align:bottom}" }{}%
7195 \ifstrequal{#1}{t}{style="\LWR@print@mbx{vertical-align:top}" }{}%
7196 }% vpos?
7197 }% rows?

```

```

7198 class="td%

```

Print the column type and vertical bars:

```

7199 \setcounter{LWR@mcolvertbarsl}{0}%
7200 \setcounter{LWR@mcolvertbarsr}{0}%
7201 \setcounter{LWR@mcolvertbarsldash}{0}%
7202 \setcounter{LWR@mcolvertbarsrdash}{0}%
7203 \setboolean{LWR@mcolvertbaronleft}{true}%
7204 \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:

```

7205 \LWR@subaddcmidruletrim%
7206 {\LWR@getexparray{LWR@trimlrules}{\arabic{LWR@tablecolindex}}}%
7207 {\LWR@getexparray{LWR@trimrrules}{\arabic{LWR@lastmulticolumn}}}%

```

Also add vertical bar class.

```

7208 \ifnumcomp{\value{LWR@mcolvertbarsl}}{=}{1}{ tvertbarl}{}%
7209 \ifnumcomp{\value{LWR@mcolvertbarsl}}{>}{1}{ tvertbarldouble}{}%
7210 \ifnumcomp{\value{LWR@mcolvertbarsr}}{=}{1}{ tvertbarr}{}%
7211 \ifnumcomp{\value{LWR@mcolvertbarsr}}{>}{1}{ tvertbarrdouble}{}%
7212 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{=}{1}{ tvertbarldash}{}%
7213 \ifnumcomp{\value{LWR@mcolvertbarsldash}}{>}{1}{ tvertbarldoubledash}{}%
7214 \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{=}{1}{ tvertbarrdash}{}%
7215 \ifnumcomp{\value{LWR@mcolvertbarsrdash}}{>}{1}{ tvertbarrdoubledash}{}%

```

Close the class tag’s opening quote: " NOT A TYPO

```

7216 %

```

```

7217 \LWR@tdstartstyles%

```

```

7218 \LWR@addtabulararrowcolor%

```

```

7219 \LWR@addcmidrulewidth%

```

```

7220 \LWR@addcdashline%

```

```

7221 \LWR@addtabularhrulecolor%

```

```

7222 \LWR@addmulticolvertrulecolor%
7223 \LWR@addformatwppalignment{#5}%
7224 \LWR@tdendstyles%
7225 }% end of the opening table data tag
7226 \global\boolfalse{LWR@intabularmetadata}%
7227 \LWR@parsemulticolumnalignment{#5}{\LWR@printmccoldata}%
7228 }

```

68.21.3 Multicolumn

`\LWR@htmlmulticolumn` $\{\langle numcols \rangle\} \{\langle alignment \rangle\} \{\langle text \rangle\}$

```

7229 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
7230 {%

```

Figure out how many extra HTML columns to add for @ and ! columns:

```

7231 \LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}

```

Create the multicolumn tag:

```

7232 \LWR@domulticolumn{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#3}%

```

Move to the next \LaTeX column:

```

7233 \addtocounter{LWR@tablecolindex}{#1}%
7234 \addtocounter{LWR@tablecolindex}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```

7235 \booltrue{LWR@skipatbang}%
7236 }

```

68.21.4 Longtable captions

longtable captions use `\multicolumn`.

`Bool LWR@starredlongtable` Per the **caption** package, step the counter if `longtable*`.

```

7237 \newbool{LWR@starredlongtable}
7238 \boolfalse{LWR@starredlongtable}

```

Per the **caption** package. User-redefinable float type.

```

7239 \providecommand*{\LTcaptype}{table}

```

`\LWR@longtabledatacaptiontag` * [*<toc entry>*] {*<caption>*}

```
7240 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
7241 {%
```

Remember the latest name for `\nameref`:

```
7242 \IfValueTF{#2}{% optional given?
7243   \ifblank{#2}% optional empty?
7244   {\LWR@setlatestname{#3}}% empty
7245   {\LWR@setlatestname{#2}}% given and non-empty
7246 }% optional given
7247 {\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:

```
7248 \LWR@tabularhtmlcolumns{1}{\arabic{LWR@tabletotalcols}}
```

Create the multicolumn tag:

```
7249 \LWR@domulticolumn{\arabic{LWR@tabletotalcols}}%
7250   {\arabic{LWR@tabhtmlcoltotal}}%
7251   {P}%
7252 {% \LWR@domulticolumn
7253 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
7254 {% yes star
7255   \LWR@figcaption%
7256   #3%
7257   \endLWR@figcaption%
7258 }%
7259 {% No star:
```

Not the star version:

Don't step the counter if `\caption[] {A caption.}`

```
7260   \ifbool{LWR@starredlongtable}%
7261   {%
7262     \ifblank{#2}% TOC entry
7263     }%
7264     {%
7265       \refstepcounter{LTcapttype}%
```

```

7266         \protected@edef\@currentlabel{%
7267             \csuse{p@LTcaption}\csuse{theLTcaption}%
7268         }%
7269     }%
7270 }{}%
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```

7271     \LWR@figcaption%
7272     \csuse{fnum@LTcaption}\CaptionSeparator#3%
7273     \endLWR@figcaption%
```

See if an optional caption was given:

```

7274     \ifblank{#2}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```

7275     {}%
```

If the optional caption was given, but might only be []:

```

7276     {% TOC entry not empty
7277         \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```

7278         {% No TOC entry
7279             \addcontentsline%
7280             {\csuse{ext@LTcaption}}%
7281             {LTcaption}%
7282             {%
7283             \protect\numberline%
7284             {\csuse{p@LTcaption}\csuse{theLTcaption}}%
7285             {\ignorespaces #3\protect\relax}%
7286             }%
7287         }% end of No TOC entry
```

The optional caption has text enclosed:

```

7288         {% yes TOC entry
7289             \addcontentsline%
7290             {\csuse{ext@LTcaption}}%
7291             {LTcaption}%
7292             {%
7293             \protect\numberline%
7294             {\csuse{p@LTcaption}\csuse{theLTcaption}}%
7295             {\ignorespaces #2\protect\relax}%
```

```

7296         }%
7297     }% end of yes TOC entry
7298 }% end of TOC entry not empty
7299 }% end of no star

```

Skip any trailing @ or ! columns for this cell:

```

7300 \booltrue{LWR@skipatbang}%
7301 }% end of \LWR@domulticolumn
7302
7303 \addtocounter{LWR@tablecolindex}{\arabic{LWR@tabletotalcols}}
7304 \addtocounter{LWR@tablecolindex}{-1}
7305
7306 }

```

68.21.5 Counting HTML tabular columns

The \LaTeX specification for a table includes a number of columns separated by the & character. These columns differ in content from line to line. Additional virtual columns may be specified by the special @ and ! columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output, @ and ! columns are placed into their own tabular columns. Thus, a \LaTeX `\multicolumn` command may span several additional @ and ! columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```

7307 \newcounter{LWR@tabhtmlcolindex}
7308 \newcounter{LWR@tabhtmlcolend}
7309 \newcounter{LWR@tabhtmlcoltotal}

```

```
\LWR@subtabularhtmlcolumns {<index>}
```

Factored from `\LWR@tabularhtmlcolumns`, which follows.

```
7310 \newcommand*{\LWR@subtabularhtmlcolumns}[1]{%
```

Temporarily define a macro equal to the @ specification for this column:

```
7311     \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{#1}}%
```

If the @ specification is not empty, add to the count:

```

7312     \ifdefempty{\LWR@atbangspec}%
7313     {}%
7314     {\addtocounter{LWR@tabhtmlcoltotal}{1}}%

```

Likewise for the ! columns:

```

7315 \edef\LWR@atbangspec{\LWR@getexpparray\LWR@colbangspec}{#1}}%
7316 \ifdefempty{\LWR@atbangspec}%
7317 {}%
7318 {\addtocounter\LWR@tabhtmlcoltotal}{1}}%
7319 }

```

`\LWR@tabularhtmlcolumns` {*(starting L^AT_EX column)*} {*(number L^AT_EX columns)*}

Compute the total number of HTML columns being spanned, considering the starting L^AT_EX table column and the number of L^AT_EX tabular columns being spanned. Any @ and ! columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```

7320 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%

```

Count the starting index, compute ending index, and begin with the count being the L^AT_EX span, to which additional @ and ! columns may be added:

```

7321 \setcounter\LWR@tabhtmlcolindex{#1}%
7322 \setcounter\LWR@tabhtmlcoltotal{#2}%
7323 \setcounter\LWR@tabhtmlcolend{#1}%
7324 \addtocounter\LWR@tabhtmlcolend{#2}%

```

If at the left edge, add the at/bang columns for the left edge:

```

7325 \ifnumcomp{\value\LWR@tabhtmlcolindex}{=}{1}{%
7326 \LWR@subtabularhtmlcolumns{leftedge}%
7327 }{}%

```

Walk across the L^AT_EX columns looking for @ and ! columns:

```

7328 \whileboolexpr{%
7329 test {%
7330 \ifnumcomp{\value\LWR@tabhtmlcolindex}{<}{\value\LWR@tabhtmlcolend}}%
7331 }%
7332 }%
7333 {%
7334 \LWR@subtabularhtmlcolumns{\arabic\LWR@tabhtmlcolindex}}%
7335 \addtocounter\LWR@tabhtmlcolindex{1}%
7336 }% whiledo
7337 }

7338 \end{warpHTML}

```

68.22 Multirow if not loaded

A default definition in case **multirow** is not loaded. This is used during table parsing.

```
7339 \begin{warpHTML}
7340 \newcommand{\multirow}[2][c]{}
7341 \end{warpHTML}
```

68.23 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 260 for the HTML versions.

for HTML & PRINT: 7342 \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](#)

`\AtBeginDocument` to adjust after the user may have loaded **multirow**, which requires several tests to determine which version is loaded and thus which options are available.

```
7343 \AtBeginDocument{
```

`\@ifundefined{@xmultirow}` determines if **multirow** was never loaded.

Null action if not loaded:

```
7344 \@ifundefined{@xmultirow}
7345 {
7346 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
7347 {+m +m +O{c} +m +O{0} +m +O{0pt} +m}%
7348 {}%
7349 }% no version of multirow was loaded
7350 {% \@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.

```

7351 % The print version:
7352 % \begin{macrocode}
7353 \ifpackageloaded{multirow}{% v2.0 or newer
7354 \ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
7355 {% v2.0+:
7356 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
7357 {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
7358 {\multicolumn{#1}{#2}{\@xmultirow[#3]{#4}{#5}{#6}{#7}{#8}}}%
7359 }
7360 {% loaded but older, probably not executed:
7361 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
7362 {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
7363 {\multicolumn{#1}{#2}{\@xmultirow[#4]{#5}{#6}{#7}{#8}}}%
7364 }
7365}% 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.

```

7366 {% v1.6 or older did not \ProvidePackage
7367 \DeclareExpandableDocumentCommand{\LWR@print@multicolumnrow}%
7368 {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
7369 {\multicolumn{#1}{#2}{\@xmultirow[#4]{#5}{#6}{#7}{#8}}}%
7370 }
7371
7372}% \@ifundefined{@xmultirow}
7373
7374 \providecommand*{\multicolumnrow}{\LWR@print@multicolumnrow}
7375
7376}% AtBeginDocument

7377 \end{warpall}

```

68.24 Utility macros inside a table

for HTML output: 7378 \begin{warpHTML}

Used to prevent opening a tabular data cell if the following token is one which does not create tabular data:

```
7379 \newcommand*{\LWR@donothing}{}

```

In case **bigdelim** is not loaded:

```
7380 \newcommand*{\ldelim}{}

```

```
7381 \newcommand*{\rdelim}{}

```

```
7382 \end{warpHTML}

```

68.25 Special-case tabular markers

for HTML & PRINT: 7383 \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 9.9.


```
7384 \newcommand*{\TabularMacro}{}

```

```
7385 \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
}
\StopDefiningTabulars

```

for HTML output: 7386 \begin{warpHTML}

```

7387 \newcommand*{\ResumeTabular}{%
7388 \global\boolfalse{LWR@exitingtabular}%
7389 \global\boolfalse{LWR@tabularmutemods}%
7390 \LWR@getmynexttoken%
7391 }

```

```
7392 \end{warpHTML}
```

for PRINT output: 7393 \begin{warpprint}

```
7394 \newcommand*{\ResumeTabular}{}%
```

```
7395 \end{warpprint}
```

68.26 Checking for a new table cell

for HTML output: 7396 \begin{warpHTML}

Bool LWR@exitingtabular When \end is found, turns off the next opening data tag.

```
7397 \newbool{LWR@exitingtabular}
```

Bool LWR@tabularmutemods Mutes HTML output for @, !, < and >.

This is used while printing the final row to generate \bottomrules.

```
7398 \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:

```
7399 \newcommand*{\LWR@tabledatacolumnntag}%
```

```
7400 {%
```

```
7401 \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:

```
7402 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
7403 \ifdefequal{\LWR@mynexttoken}{\end}%
```

```
7404 {\global\booltrue{LWR@exitingtabular}}{}%
```

longtable can have a caption in a cell

```
7405 \ifdefequal{\LWR@mynexttoken}{\caption}%
```

```
7406 {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```

7407 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
7408     {\let\mynext\LWR@donothing}{}%
7409 \ifdefequal{\LWR@mynexttoken}{\multirow}%
7410     {\let\mynext\LWR@donothing}{}%
7411 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
7412     {\let\mynext\LWR@donothing}{}%
7413 \ifdefequal{\LWR@mynexttoken}{\noalign}%
7414     {\let\mynext\LWR@donothing}{}%

```

If an `\mrowcell`, this is a cell to be skipped over:

```

7415 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
7416     {\let\mynext\LWR@donothing}{}%

```

If an `\mcolrowcell`, this is a cell to be skipped over:

```

7417 \ifdefequal{\LWR@mynexttoken}{\mcolrowcell}%
7418     {\let\mynext\LWR@donothing}{}%

```

```

7419 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
7420     {\let\mynext\LWR@donothing}{}%

```

```

7421 \ifdefequal{\LWR@mynexttoken}{\hline}%
7422     {\let\mynext\LWR@donothing}{}%

```

```

7423 \ifdefequal{\LWR@mynexttoken}{\firsthline}%
7424     {\let\mynext\LWR@donothing}{}%

```

```

7425 \ifdefequal{\LWR@mynexttoken}{\lasthline}%
7426     {\let\mynext\LWR@donothing}{}%

```

```

7427 \ifdefequal{\LWR@mynexttoken}{\toprule}%
7428     {\let\mynext\LWR@donothing}{}%

```

```

7429 \ifdefequal{\LWR@mynexttoken}{\midrule}%
7430     {\let\mynext\LWR@donothing}{}%

```

```

7431 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
7432     {\let\mynext\LWR@donothing}{}%

```

```

7433 \ifdefequal{\LWR@mynexttoken}{\morecmidrules}%
7434     {\let\mynext\LWR@donothing}{}%

```

```

7435 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
7436     {\let\mynext\LWR@donothing}{}%

```

```

7437 \ifdefequal{\LWR@mynexttoken}{\cline}%
7438     {\let\mynext\LWR@donothing}{}%

7439 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
7440     {\let\mynext\LWR@donothing}{}%

7441 \ifdefequal{\LWR@mynexttoken}{\rowcolor}%
7442     {\let\mynext\LWR@donothing}{}%

7443 \ifdefequal{\LWR@mynexttoken}{\arrayrulecolor}%
7444     {\let\mynext\LWR@donothing}{}%

7445 \ifdefequal{\LWR@mynexttoken}{\doublerulesepcolor}%
7446     {\let\mynext\LWR@donothing}{}%

7447 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
7448     {\let\mynext\LWR@donothing}{}%

7449 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
7450     {\let\mynext\LWR@donothing}{}%

7451 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
7452     {\let\mynext\LWR@donothing}{}%

7453 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
7454     {\let\mynext\LWR@donothing}{}%

```

For `arydshln`:

```

7455 \ifdefequal{\LWR@mynexttoken}{\hdashline}%
7456     {\let\mynext\LWR@donothing}{}%

7457 \ifdefequal{\LWR@mynexttoken}{\cdashline}%
7458     {\let\mynext\LWR@donothing}{}%

7459 \ifdefequal{\LWR@mynexttoken}{\firsthdashline}%
7460     {\let\mynext\LWR@donothing}{}%

7461 \ifdefequal{\LWR@mynexttoken}{\lasthdashline}%
7462     {\let\mynext\LWR@donothing}{}%

```

Ignore an empty line between rows:

```

7463 \ifdefequal{\LWR@mynexttoken}{\par}%
7464     {\let\mynext\LWR@donothing}{}%

```

No action for an `\end` token.

Add similar to the above for any other non-data tokens which might appear in the table.

Start the new table cell if was not any of the above:

```


7465 \LWR@traceinfo{LWR@tabledatacolumnntag: about to do mynext}%
7466 \mynext%
7467 \LWR@traceinfo{LWR@tabledatacolumnntag: done}%
7468 }

7469 \end{warpHTML}

```

68.27 `\mrowcell`

for HTML & PRINT: 7470 `\begin{warpall}`

`\mrowcell`  The user must insert `\mrowcell` into any `\multirow` cells which must be skipped. This command has no action during print output.

```


7471 \newcommand*{\mrowcell}{}

7472 \end{warpall}

```

68.28 `\mcolrowcell`

for HTML & PRINT: 7473 `\begin{warpall}`

`\mcolrowcell`  The user must insert `\mcolrowcell` into any `\multicolumnrow` cells which must be skipped. This command has no action during print output.

```

7474 \newcommand*{\mcolrowcell}{}

7475 \end{warpall}

```

68.29 HTML tabular environment

for HTML output: 7476 `\begin{warpHTML}`

These are default definitions in case **booktabs** is not loaded, and are not expected to be used, but must exist as placeholders.

```

7477 \newcommand*{\toprule}[1] [] {\hline}
7478 \newcommand*{\midrule}[1] [] {\hline}
7479 \LetLtxMacro\cmidrule\cline
7480 \newcommand*{\bottomrule}[1] [] {\hline}
7481 \newcommand*{\addlinespace}[1] [] {}
7482 \newcommand*{\morecmidrules}{}
7483 \newcommand*{\specialrule}[3] {\hline}

```

`\noalign` `{\text{}}` Redefined for use inside tabular.

```

7484 \LetLtxMacro\LWR@orignoalign\noalign
7485
7486 \newcommand{\LWR@tabularnoalign}[1]{%
7487 \begingroup%
7488 \global\advance\rownum\m@ne%
7489 \renewcommand*{\LWR@xcolorrowHTMLcolor}{}%
7490 \multicolumn{\value{\LWR@tabletotalcols}}{1}{#1} \\\
7491 \endgroup%
7492 % \@rowcolors%
7493 \LWR@getmynexttoken%
7494 }

```

`\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.

```

7495 \AtBeginDocument{
7496 \@ifpackageloaded{lwarp-tabls}
7497 {
7498 \newcommand*{\LWR@HTMLhline}[1] [] {%
7499   \ifbool{FormatWP}%
7500   {\LWR@docmidrule{1-\arabic{\LWR@tabletotalcols}}}%
7501   {\addtocounter{\LWR@hlines}{1}}%
7502   \LWR@getmynexttoken}%
7503 }
7504 {
7505 \newcommand*{\LWR@HTMLhline}{%
7506   \ifbool{FormatWP}%
7507   {\LWR@docmidrule{1-\arabic{\LWR@tabletotalcols}}}%
7508   {\addtocounter{\LWR@hlines}{1}}%
7509   \LWR@getmynexttoken}%
7510 }
7511 }% \AtBeginDocument

```

`\LWR@HTMLcline` $\{\langle columns \rangle\}$

```
7512 \NewDocumentCommand{\LWR@HTMLcline}{m}%
7513 {\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**.

```
7514 \AtBeginDocument{
7515 \ifundefined{frenchbsetup}%
7516 {% no babel-french
7517   \newcommand*\LWR@nullifyNoAutoSpacing{}
7518 }% no babel-french
7519 {% yes babel-french
7520   \newcommand*\LWR@nullifyNoAutoSpacing{%
7521     \NoAutoSpacing%
7522     \renewcommand*\NoAutoSpacing{}%
7523     \renewcommand*\LWR@FBcancel{}%
7524   }
7525 }% yes babel-french
7526 }% 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.

```
7527 \StartDefiningTabulars
7528
7529 \newenvironment*LWR@tabular}[2] []
7530 {%
7531 \LWR@traceinfo{LWR@tabular started}%
7532 \addtocounter{LWR@tabulardepth}{1}%
```

Not yet started a table row:

```
7533 \global\boolfalse{LWR@startedrow}%
```

Not yet doing any rules:

```
7534 \setcounter{LWR@hlines}{0}%
7535 \setcounter{LWR@hdashedlines}{0}%
7536 \global\boolfalse{LWR@doingtbrule}%
7537 \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.

```
7538 \LWR@nullifyNoAutoSpacing%
```

Have not yet found the end of tabular command. Unmute the @ and ! columns.

```
7539 \global\boolfalse{LWR@exitingtabular}%
7540 \global\boolfalse{LWR@tabularmutemods}%
```

Create the table tag:

```
7541 \global\booltrue{LWR@intabularmetadata}%
7542 \LWR@traceinfo{LWR@tabular: About to LWR@forecenewpage.}%
7543 \LWR@forcenewpage
7544 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
7545 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of llccrr, etc.

Do not place the table inside a paragraph:

```
7546 \LWR@stoppars%
```

Track column #:

```
7547 \setcounter{LWR@tablecolindex}{1}%
```

Have not yet added data in this column:

```
7548 \boolfalse{LWR@tabularcelladded}%
```

Start looking for midrules:

```
7549 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
7550 \LetLtxMacro{\\}{LWR@tabularendoffline}%
```

The following adjust for **colortbl**.

```
7551 \LetLtxMacro\arrayrulecolor\arrayrulecolornexttoken%
7552 \LetLtxMacro\doublerulesepcolor\doublerulesepcolornexttoken%
```

```

7553 \renewcommand*{\LWR@columnHTMLcolor}{}%
7554 \renewcommand*{\LWR@rowHTMLcolor}{}%
7555 \renewcommand*{\LWR@cellHTMLcolor}{}%
7556 \@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.

```

7557 \edef\LWR@vertruleHTMLcolor{\LWR@ruleHTMLcolor}%

```

Tracking the depth of cell color <div>s:

```

7558 \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 **tabls** package.

```

7559 \LWR@traceinfo{LWR@tabular: redefining macros}%
7560 \LetLtxMacro\noalign\LWR@tabularnoalign%
7561 \LetLtxMacro\hline\LWR@HTMLhline%
7562 \LetLtxMacro\cline\LWR@HTMLcline%

7563 \DeclareDocumentCommand{\hdashline}{o}{%
7564   \ifbool{FormatWP}%
7565     {\LWR@docdashline{1-\arabic{LWR@tabletotalcols}}}%
7566     {\addtocounter{LWR@hdashedlines}{1}}%
7567   \LWR@getmynexttoken%
7568 }%

7569 \DeclareDocumentCommand{\cdashline}{m}{%
7570   \LWR@docdashline{##1}\LWR@getmynexttoken%
7571 }%

7572 \DeclareDocumentCommand{\firsthdashline}{o}{%
7573   \ifbool{FormatWP}%
7574     {\LWR@docdashline{1-\arabic{LWR@tabletotalcols}}}%
7575     {\addtocounter{LWR@hdashedlines}{1}}%
7576   \LWR@getmynexttoken%
7577 }%

7578 \DeclareDocumentCommand{\lasthdashline}{o}{%
7579   \ifbool{FormatWP}%
7580     {\LWR@docdashline{1-\arabic{LWR@tabletotalcols}}}%
7581     {\addtocounter{LWR@hdashedlines}{1}}%
7582   \LWR@getmynexttoken%

```

7583 }%

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.

```
7584 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
7585 \renewcommand*{\mrowcell}{%
7586     \LWR@maybenewtablerow%
7587     \LWR@tabularleftedge%
7588     \global\booltrue{\LWR@skippingmrowcell}%
7589 }%
7590 \renewcommand*{\mcolrowcell}{%
7591     \LWR@maybenewtablerow%
7592     \global\booltrue{\LWR@skippingmcolrowcell}%
7593 }%
7594 \LetLtxMacro\caption\LWR@longtabledatacaptiontag%
```

Reset for new processing:

```
7595 \global\boolfalse{\LWR@tableparcell}%
7596 \global\boolfalse{\LWR@skippingmrowcell}%
7597 \global\boolfalse{\LWR@skippingmcolrowcell}%
7598 \global\boolfalse{\LWR@skipatbang}%
7599 \global\boolfalse{\LWR@emptyatbang}%

```

Set & for its special meaning inside the tabular:

```
7600 \StartDefiningTabulars%
7601 \protected\gdef&{\LWR@tabularampersand}%

```

Nest one level deeper of tabular paragraph handling:

```
7602 \addtocounter{\LWR@tabularpardepth}{1}%

```

Look ahead for a possible table data cell:

```
7603 \LWR@traceinfo{\LWR@tabular: about to \LWR@getmynexttoken}%
7604 \LWR@getmynexttoken%
7605 }%

```

Ending the environment:

```
7606 {%
7607 \LWR@traceinfo{\LWR@tabular ending}%

```

Unnest one level of tabular paragraph handling:

```
7608 \addtocounter{\LWR@tabularpardepth}{-1}%

```

```

7609 \ifboolexpr{%
7610     test {%
7611         \ifnumcomp{\value{LWR@tablecolindex}}{<}{\value{LWR@tabletotalcols}}
7612     } or %
7613     (%
7614         bool{LWR@intabularmetadata} and%
7615         not bool{LWR@tabularcelladded} and%
7616         test {%
7617             \ifnumcomp{\value{LWR@tablecolindex}}{=}{\value{LWR@tabletotalcols}}%
7618         }%
7619     )%
7620 }%
7621 {%
7622     \LWR@tabularfinishrow%
7623 }%
7624 {%
7625     \LWR@closetabledatacell%
7626 }%
7627 \LWR@htmlblocktag{/tr}%

```

xcolor row color support:

```

7628 \@rowcolor%

7629 \LWR@htmlblocktag{/table}%
7630 \global\boolfalse{LWR@intabularmetadata}%

```

Unnest one level of tabular:

```

7631 \addtocounter{LWR@tabulardepth}{-1}%

```

Restore & to its usual meaning:

```

7632 \protected\gdef&{\LWR@origampmacro}%
7633 \StopDefiningTabulars%
7634 \LWR@traceinfo{LWR@tabular finished ending}%
7635 }
7636
7637 \StopDefiningTabulars

7638 \end{warpHTML}

```

69 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The **zref** package is used to remember section name, file, and lateximage depth and number for each label.

Table 10 shows the data structures related to cross-referencing.

for HTML output: 7639 \begin{warpHTML}

69.1 Setup

\@currentlabelname To remember the most recently defined section name, description, or caption, for \nameref.

```
7640 \providecommand*{\@currentlabelname}{}%
```

\LWR@stripperperiod {\text} [(.)]

Removes a trailing period.

```
7641 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

\LWR@setlatestname {\object name}

Removes \label, strips any final period, and remembers the result.

```
7642 \newcommand*{\LWR@setlatestname}[1]{%
```

Remove \label and other commands from the name, the strip any final period. See **zref-titleref** and **getttitlestring**.

```
7643 \GetTitleStringExpand{#1}%
7644 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
7645 \edef\@currentlabelname{%
7646 \expandafter\LWR@stripperperiod\@currentlabelname%
7647 \ltx@empty.\ltx@empty\@nil%
7648 }%
7649 }
```

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 69.3), plus <code>\splabel</code> data (section 69.2): <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 69.4. <code>\ref</code> and <code>\nameref:</code> Adds HTML tags. See section 69.4.	
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 55.4 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 83. <code>\label inside math:</code> See section 75.7.1.	
Footnotes: See <code>\noteentry</code> in section 55.4.	

69.2 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:

```
7650 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
7651 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
7652 \zref@newprop{zLWR@htmlfilenumber}{%
7653 \ifbool{FileSectionNames}{\LWR@thisfilename}{\arabic{LWR@htmlfilenumber}}}%
7654 }%
```

Additional properties for lateximages:

```
7655 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
7656 \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:

```
7657 \zref@addprop{special}{zLWR@name}
7658 \zref@addprop{special}{zLWR@htmlfilenumber}
7659 \zref@addprop{special}{zLWR@lateximagedepth}
7660 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
7661 \newcommand*{\LWR@spref}[2]{%
7662 \zref@extractdefault{#1}{#2}{??}%
7663 }
```

`\LWR@nameref` `{\langle label \rangle}` Returns the section name for this label:

```
7664 \newcommand*{\LWR@nameref}[1]{%
7665 \LWR@spref{#1}{zLWR@name}%
7666 }
```

`\LWR@htmlfileref` $\{\langle label \rangle\}$ Returns the file number or name for this label:

```
7667 \newcommand*\LWR@htmlfileref}[1]{%
```

DO NOT USE `\LWR@traceinfo` HERE! Will be expanded.

```
7668 \LWR@spref{#1}{z\LWR@htmlfilenumber}%
7669 }
```

`\LWR@lateximagedepthref` $\{\langle label \rangle\}$ Returns the `lateximagedepth` for this label:

```
7670 \newcommand*\LWR@lateximagedepthref}[1]{%
7671 \LWR@spref{#1}{z\LWR@lateximagedepth}%
7672 }
```

`\LWR@lateximagenumberref` $\{\langle label \rangle\}$ Returns the `lateximagenumber` for this label:

```
7673 \newcommand*\LWR@lateximagenumberref}[1]{%
7674 \LWR@spref{#1}{z\LWR@lateximagenumber}%
7675 }
```

`\LWR@splabel` $\{\langle label \rangle\}$ Sanitize the name and then creates the label:

```
7676 \newcommand*\LWR@splabel}[1]{%
7677 \LWR@traceinfo{LWR@splabel !#1!}%
7678 \LWR@setlatestname{\@currentlabelname}%
7679 \zref@labelbylist{#1}{special}%
7680 }
```

69.3 Labels

`\LWR@sublabel` $\{\langle label \rangle\}$ Creates an HTML id tag.

`\detokenize` is used to allow underscores in the labels.

```
7681 \newcommand*\LWR@sublabel}[1]{%
7682 \LWR@traceinfo{LWR@sublabel !#1!}%
```

Create an HTML id tag unless are inside a `lateximage`, since it would appear in the image:

```
7683 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
7684 {}%
7685 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```

7686 \LWR@sanitize{#1}%
7687 \ifbool{LWR@doingstartpars}%
7688 {% pars allowed
7689 \ifbool{LWR@doingapar}%
7690 {% par started
7691 \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
7692 }% par started
7693 {% par not started
7694 \LWR@stoppars%
7695 \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
7696 \LWR@startpars%
7697 }% par not started
7698 }% pars allowed
7699 {% pars not allowed
7700 \LWR@htmltag{a \LWR@print@mbox{id="\LWR@sanitized"}}\LWR@htmltag{/a}%
7701 }% pars not allowed
7702 }% not lateximage
7703 }
```

`\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.

```

7704 \NewDocumentCommand{\LWR@newlabel}{d() m o}{%
7705 \LWR@traceinfo{LWR@newlabel: starting}%
7706 \LWR@traceinfo{LWR@newlabel: !#2!}%
7707 % \@bsphack%
```

Create a traditional \LaTeX label, as modified by **cleveref**:

```

7708 \LWR@origlabel{#2}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

7709 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
7710 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
7711 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \arabic{LWR@htmlfilenumber}}%
7712 \LWR@splabel{#2}%
7713 \LWR@sublabel{#2}%
```

```

7714 % \@esphack%
7715 \LWR@traceinfo{LWR@newlabel: done}%
7716 }

```

69.4 References

`\LWR@startref` `{\label}` (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```

7717 \newcommand*{\LWR@startref}[1]
7718 {%
7719 \edef\LWR@lhref{\LWR@lateximagedepthref{#1}}%
7720 \LWR@sanitize{#1}%
7721 \LWR@traceinfo{LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

7722 \LWR@htmltag{a href="%
7723 \LWR@traceinfo{LWR@startref B}%
7724 \LWR@print@mbox{\LWR@htmlrefsectionfilename{#1}}%
7725 \LWR@traceinfo{LWR@startref C}%
7726 \LWR@origpound%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

7727 \LWR@traceinfo{LWR@startref D: !#1!}%
7728 \ifthenelse{\equal{\LWR@lhref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

7729 {%
7730 \LWR@traceinfo{LWR@startref D0: ??}%
7731 ??%
7732 }%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

7733 {%
7734 \LWR@traceinfo{LWR@startref D1: \LWR@lhref}%
7735 \ifthenelse{\cinttest{\LWR@lhref}{>}{0}}%

```

```

7736    {%
7737        \LWR@traceinfo{\LWR@startref D2: \LWR@lidenref}%
7738        lateximage\LWR@lateximagenumberref{#1}%
7739    }%
7740    {%
7741        \LWR@traceinfo{\LWR@startref D3}%

```

`\detokenize` is used to allow underscores in the labels:

```

7742        \LWR@print@mbbox{\LWR@sanitized}%
7743    }%
7744}%
7745 \LWR@traceinfo{\LWR@startref E}%

```

Closing quote:

```

7746 "%
7747 \LWR@traceinfo{\LWR@startref F}%
7748 }

```

`\LWR@subnewref` *{<label>} {<label or sub@label>}*

Factored for the **subfig** package. Uses the original label for the hyper-reference, but prints its own text, such as “1(b)”.

```

7749 \NewDocumentCommand{\LWR@subnewref}{m m}{%
7750 \LWR@traceinfo{\LWR@subnewref #1 #2}%
7751 \LWR@startref{#1}%
7752 \LWR@origref{#2}%
7753 \LWR@htmltag{/a}%
7754 }

```

`\ref` * *{<label>}* `\ref` is redefined to `\LWR@newref`, except inside the text part of a `\hyperref`, where it is redefined to `\LWR@ref@ignorestar`.

`\LWR@newref` * *{<label>}* Create an internal document reference link, or without a link if starred per **hyperref**.

```

7755 \NewDocumentCommand{\LWR@newref}{s m}{%
7756 \LWR@traceinfo{\LWR@newref !#2!}%
7757 \IfBooleanTF{#1}%
7758 {\LWR@origref{#2}}%
7759 {\LWR@subnewref{#2}{#2}}%
7760 }

```

`\LWR@ref@ignorestar` * *{<label>}* For use inside `\hyperref`. Ignores the star, then uses the original `\ref`.

```

7761 \NewDocumentCommand{\LWR@ref@ignorestar}{s m}{%
7762     \LWR@origref{#2}%
7763 }

```

`\pagerefPageFor` Text for page references.

```

7764 \newcommand*{\pagerefPageFor}{see }

```

`\pageref` * `{\label}` Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```

7765 \NewDocumentCommand{\LWR@newpageref}{s m}{%
7766 \IfBooleanTF{#1}%
7767 {(\pagerefPageFor\LWR@origref{#2})}%
7768 {(\cpageref{#2})}%
7769 }

```

`\nameref` `{\label}`

```

7770 \newrobustcmd*{\nameref}[1]{%
7771 \LWR@traceinfo{nameref}%
7772 \LWR@startref{#1}%
7773 \LWR@traceinfo{nameref B}%
7774 \LWR@nameref{#1}%
7775 \LWR@traceinfo{nameref C}%
7776 \LWR@htmltag{/a}%
7777 \LWR@traceinfo{nameref: done}%
7778 }

```

`\Nameref` `{\label}` In print, adds the page number. In HTML, does not.

```

7779 \LetLtxMacro\Nameref\nameref

```

69.5 Hyper-references



Note that the code currently only sanitizes the underscore character. Additional characters should be rendered inert as well. See the `hyperref.sty` definition of `\gdef\hyper@normalise` for an example.

Pkg `hyperref`



Do not tell other packages that `hyperref` is emulated. Some packages patch various

commands if **hyperref** is present, which will probably break something, and the emulation already handles whatever may be emulated anyhow.



Any reference to `\usepackage{hyperref}` must be placed inside a `warpprint` environment.

```
7780 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
7781 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Emulates **hyperref**:

`\@currentHref` Added to support **backref**.

```
7782 \AtBeginDocument{
7783 \def\@currentHref{%
7784 autopage-\theLWR@currentautosec%
7785 }
7786 }
```

`\LWR@subhyperref` `{\langle URL \rangle}`

Starts a link for `\LWR@hrefb`. A group must have been opened first, with nullified catcodes. The text name is printed afterwards, after the group is closed and catcodes restored.

```
7787 \NewDocumentCommand{\LWR@subhyperref}{m}{%
7788 \LWR@traceinfo{\LWR@subhyperref !#1!}%
7789 \LWR@sanitize{#1}%
7790 \LWR@htmltag{%
7791     a href="\LWR@sanitized" %
7792     target="\_{}blank"\LWR@orignewline%
7793 }%
7794 }
```

`\LWR@subhyperreftext` `{\langle text \rangle}`

Finishes the **hyperref** for `\LWR@hrefb`. Catcodes must have been restored already. To be used after `\LWR@subhyperref`, and after its group has been closed.

```
7795 \newcommand{\LWR@subhyperreftext}[1]{%
7796 #1%
7797 \LWR@htmltag{/a}%
7798 \LWR@ensuredoingapar%
7799 }
```

`\LWR@subhyperrefclass` `{\langle URL \rangle} {\langle text \rangle} {\langle htmlclass \rangle}`

```

7800 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
7801 \LWR@htmltag{%
7802     a href="%
7803         \begingroup\@sanitize#1\endgroup%
7804     " %
7805     class="#3"\LWR@orignewline%
7806 }%
7807 #2%
7808 \LWR@htmltag{/a}%
7809 \LWR@ensuredoingapar%
7810 }

```

`\href` [*options*] {*URL*}

Create a link with accompanying text:

```

7811 \DeclareDocumentCommand{\LWR@hrefb}{0{} m}{%
7812 \LWR@ensuredoingapar%
7813 \LWR@subhyperref{#2}%
7814 \endgroup% restore catcodes
7815 \LWR@subhyperreftext%
7816 }
7817
7818 \newrobustcmd*{\href}{%
7819 \begingroup%
7820 \catcode'\#=12%
7821 \catcode'\%=12%
7822 \catcode'\&=12%
7823 \catcode'\~=12%
7824 \catcode'\_ =12%
7825 \LWR@hrefb%
7826 }

```

`\nolinkurl` {*URL*}

Print the name of the link without creating the link:

```

7827 \newcommand*{\LWR@nolinkurlb}[1]{%
7828 \LWR@ensuredoingapar%
7829 \def\LWR@templink{#1}%
7830 \@onelevel@sanitize\LWR@templink%
7831 \LWR@templink%
7832 \endgroup%
7833 }
7834
7835 \newrobustcmd*{\nolinkurl}{%
7836 \begingroup%
7837 \catcode'\#=12%

```

```

7838 \catcode'\%=12%
7839 \catcode'\&=12%
7840 \catcode'\~=12%
7841 \catcode'\_ =12%
7842 \LWR@nolinkurlb%
7843 }

```

`\url` $\{\langle URL \rangle\}$

Create a link whose text name is the address of the link.

The `url` package may redefine `\url`, so it is `\let to \LWR@urlahere` and also redefined by `lwarp-url`.

```

7844 \DeclareDocumentCommand{\LWR@urlb}{m}{%
7845 \LWR@ensuredoingapar%
7846 \def\LWR@templink{#1}%
7847 \@onelevel@sanitize\LWR@templink%
7848 \href{\LWR@templink}{\LWR@templink}%
7849 \endgroup%
7850 }
7851
7852 \newrobustcmd*{\url}{%
7853 \begingroup%
7854 \catcode'\#=12%
7855 \catcode'\%=12%
7856 \catcode'\&=12%
7857 \catcode'\~=12%
7858 \catcode'\_ =12%
7859 \LWR@urlb%
7860 }

```

`\LWR@subinlineimage` $[\langle alttag \rangle] \{\langle class \rangle\} \{\langle filename \rangle\} \{\langle extension \rangle\} \{\langle style \rangle\}$

```

7861 \newcommand*{\LWR@subinlineimage}[5] [] {%
7862 \ifblank{#1}%
7863 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"}}%
7864 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"}}%
7865 }
7866 \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.
-

70 Floats

Floats are supported, although partially through emulation.

Table 11 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the **babel** package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

70.1 Float environment

for HTML output: 7867 \begin{warpHTML}

\LWR@floatbegin {<type>} [<placement>] Begins a \newfloat environment.

```
7868 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
7869 \ifbool{FormatWP}{\newline}{}}%
7870 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
7871 \addtocounter{LWR@thisautoid}{1}%
7872 \booltrue{LWR@freezethisautoid}%
```

```
7873 \begingroup%
```

Settings while inside the environment:

```
7874 \LWR@origraggedright%
```

Open an HTML figure tag. The figure is assigned a class equal to its type, and another class according to the **float** package style, if used. Note that \csuse returns an empty string if \LWR@floatstyle@<type> is not defined.

```
7875 \LWR@htmltag{%
7876   figure id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}" % space
7877   class="#1 \csuse{LWR@floatstyle@#1}"%
7878 }%
7879 \ifbool{FormatWP}{%
7880   \LWR@orignewline%
7881   \LWR@BlockClassWP}{-}{wp#1}%
7882 }{}%
```

Update the caption type:

```
7883 \renewcommand*{\@capttype}{#1}%
7884 \caption@settype{#1}%
```

Mark the float for a word processor conversion:

```
7885 \LWR@startpars%
7886 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
7887
7888 === begin #1 ===
7889
7890 }{}%
```

Look for `\centering`, etc:

```
7891 \LWR@futurenonspacel\LWR@mynexttoken\LWR@floatalignment%
7892 }
```

For **koma-script**. The following does not work for tables.

```
7893 \AtBeginDocument{
7894 \ifpackageloaded{tocbasic}{
7895 \appto\figure@atbegin{%
7896 \LWR@futurenonspacel\LWR@mynexttoken\LWR@floatalignment%
7897 }
7898 }{}
7899 }
```

`\@float` Support packages which create floats directly.
`\@dblfloat`

```
7900 \let\@float\LWR@floatbegin
7901 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
7902 \newcommand*{\LWR@floatend}{%
```

If saw a `\centering`, finish the center environment:

```
7903 \LWR@endfloatalignment%
```

Mark the float end for a word processor conversion:

```
7904 \ifboolexpr{bool{FormatWP} and bool{WPMarkFloats}}{%
7905
7906 === end ===
7907
7908 }{}%
7909 \LWR@stoppars%
```

Close an HTML figure tag:

```
7910 \ifbool{FormatWP}{\endLWR@BlockClassWP}{}%
7911 \LWR@html\elementend{figure}%
7912 \endgroup%
7913 \boolfalse{LWR@freezethisautoid}%
7914 \LWR@startpars%
7915 \ifbool{FormatWP}{\newline}{}%
7916 }
```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

7917 `\let\end@float\LWR@floatend`
 7918 `\let\end@dblfloat\LWR@floatend`

70.2 Float tracking

Ctrl `LWR@thisautoid` A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

7919 `\newcounter{LWR@thisautoid}`

Ctrl `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.

7920 `\newcounter{LWR@thisautoidWP}`

Bool `LWR@freezethisautoid` Prevents multiple increments of `\LWR@thisautoid` inside a float.

7921 `\newbool{LWR@freezethisautoid}`
 7922 `\boolfalse{LWR@freezethisautoid}`

`\LWR@newautoidanchor` Adds a new `<autoid>` anchor.

7923 `\newcommand*{\LWR@newautoidanchor}{%`
 7924 `\ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%`
 7925 `{}}%`
 7926 `{`
 7927 `\ifbool{LWR@freezethisautoid}{}{%`
 7928 `\addtocounter{LWR@thisautoid}{1}%`
 7929 `\LWR@htmltag{a id="\LWR@print@mbox{autoid-\arabic{LWR@thisautoid}}"}%`
 7930 `\LWR@htmltag{/a}%`
 7931 `}%`
 7932 `}`
 7933 `}`

`\@capttype` Remembers which float type is in use.

7934 `\newcommand*{\@capttype}{}%`

`\LWR@floatalignmentname` Set to center, flushleft, or flushright if saw `\centering`, `\raggedright`, or `\raggedleft`.

7935 `\newcommand*{\LWR@floatalignmentname}{}%`

`\LWR@floatalignment` If sees a `\centering`, `\raggedleft`, or `\raggedright`, creates a center, flushright, or flushleft environment.

```

7936 \newcommand*{\LWR@floatalignment}{%
7937 \ifdefstrequal{\LWR@mynexttoken}{\centering}{%
7938     \center%
7939     \renewcommand*{\LWR@floatalignmentname}{center}%
7940 }{}%
7941 \ifdefstrequal{\LWR@mynexttoken}{\raggedright}{%
7942     \flushleft%
7943     \renewcommand*{\LWR@floatalignmentname}{flushleft}%
7944 }{}%
7945 \ifdefstrequal{\LWR@mynexttoken}{\raggedleft}{%
7946     \flushright%
7947     \renewcommand*{\LWR@floatalignmentname}{flushright}%
7948 }{}%
7949 }
```

`\LWR@endfloatalignment` Closes an environment from `\LWR@floatalignment`.

```

7950 \newcommand*{\LWR@endfloatalignment}{%
7951 \ifdefvoid{\LWR@floatalignmentname}{}\csuse{end\LWR@floatalignmentname}}%
7952 \renewcommand*{\LWR@floatalignmentname}{}%
7953 }
```

70.3 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

7954 \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.

```

7955 \AtBeginDocument{\renewcommand{\@makecaption}[2]{%
7956     \LWR@traceinfo{\@makecaption}%
7957     #1\CaptionSeparator#2%
7958     \LWR@traceinfo{\@makecaption: done}%
7959 }%
7960 }
```

70.4 Caption and LOF linking and tracking

When a new HTML file is marked in the \LaTeX PDF file, the \LaTeX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special \LaTeX labels). This page number is used to generate an `autopage HTML <id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an `HTML autoid <id>` at the start of the float itself in the HTML file. The `autopage` and `autoid` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

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.

```
7961 \newcounter{LWR@nextautoid}
7962 \newcounter{LWR@nextautopage}
```

`\LWRsetnextfloat` `{\<autopage>}{\<autoid>}`

This is written to the `.lof`, `.lot` file just before each float's usual entry. The `autopage` and `autoid` are remembered for `\l@figure` to use when creating the HTML links.

```
7963 \newcommand*{\LWRsetnextfloat}[2]{%
7964 \setcounter{LWR@nextautopage}{#1}%
7965 \setcounter{LWR@nextautoid}{#2}%
7966 }
```

Ctrl `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the `autoid` by the modified `\addcontentsline` just before each float's entry.

```
7967 \newcounter{LWR@latestautopage}
7968 \setcounter{LWR@latestautopage}{1}
```

Env `LWR@figcaption` Encapsulates a caption inside `<figcaption>`, and if `FormatWP` then also a `<div>` with an italic style.

```
7969 \newenvironment*{LWR@figcaption}
7970 {%
7971 \LWR@traceinfo{LWR@figcaption env start}%
7972 \LWR@htmlblocktag{figcaption}%
7973 \ifbool{FormatWP}{%
7974 \begin{BlockClass}[font-style:italic]{italic}
7975 \LWR@print@vspace*{\baselineskip}
7976 }{}%
7977 \LWR@traceinfo{LWR@figcaption env start: done}%
```

```

7978 }
7979 {%
7980 \LWR@traceinfo{LWR@figcaption env end}%
7981 \ifbool{FormatWP}{\end{BlockClass}}{}%
7982 \LWR@htmlblocktag{/figcaption}%
7983 \LWR@traceinfo{LWR@figcaption env end: done}%
7984 }

```

After packages have loaded, remember the print-mode version of the following:

```

7985 \AtBeginDocument{
7986 \LetLtxMacro\LWR@origcaption@begin\caption@begin
7987 \LetLtxMacro\LWR@origcaption@end\caption@end
7988 }

```

`\LWR@caption@begin` `{\langle type \rangle}`

Low-level patches to create HTML tags for captions.

```

7989 \newcommand{\LWR@caption@begin}[1]
7990 {
7991 \LWR@traceinfo{LWR@caption@begin}%

```

Keep par and minipage changes local:

```

7992 \begingroup%

```

The **caption** code was not allowing the closing par tag:

```

7993 \@setpar{\LWR@closeparagraph\@@par}%

```

No need for a minipage or `\parbox` inside the caption:

```

7994 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
7995 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%

```

Enclose the original caption code inside an HTML tag:

```

7996 \LWR@figcaption%
7997 \LWR@traceinfo{LWR@caption@begin: about to LWR@origcaption@begin}%
7998 \LWR@origcaption@begin{#1}%
7999 \LWR@traceinfo{LWR@caption@begin: done}%
8000 }

```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```

8001 \newcommand{\LWR@caption@end}
8002 {%
8003 \LWR@traceinfo{\LWR@caption@end}%
8004 \LWR@origcaption@end%

```

Closing tag:

```

8005 \endLWR@figcaption%
8006 \endgroup%
8007 % \leavevmode% avoid bad space factor (0) error
8008 \LWR@traceinfo{\LWR@caption@end: done}%
8009 }

```

`\caption@begin` Low-level patches to create HTML tags for captions.
`\caption@end`

```

8010 \AtBeginDocument{
8011 \let\caption@begin\LWR@caption@begin
8012 \let\caption@end\LWR@caption@end
8013 }

```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```

8014 \let\LWR@origcaptionlistentry\captionlistentry
8015
8016 \renewcommand*{\captionlistentry}{%
8017 \LWR@ensuredoingapar%
8018 \LWR@origcaptionlistentry%
8019 }
8020
8021 \def\LWR@LTcaptionlistentry{%
8022 \LWR@ensuredoingapar%
8023 \LWR@htmltag{a id="\LWR@print@mbox{autoid-\arabic{\LWR@thisautoid}}"}\LWR@htmltag{/a}%
8024 \bgroup
8025 \@ifstar{\egroup\LWR@LT@captionlistentry}% gobble *
8026 {\egroup\LWR@LT@captionlistentry}}%
8027 \def\LWR@LT@captionlistentry#1{%
8028 \caption@listentry\@firstoftwo[\LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autoid before each float's entry. No changes if writing .toc For a theorem, automatically defines `\ext@<type>` as needed, to mimic and reuse the float mechanism.

f

```

8029 \let\LWR@origaddcontentsline\addcontentsline
8030

```

```

8031 \renewcommand*{\addcontentsline}[3]{%
8032 \ifstrequal{#1}{toc}{\% not TOC

8033 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
8034     }%
8035     {\LWR@newautoidanchor}%

8036     \ifcsvoid{ext@#2}{\csdef{ext@#2}{#1}}{%
8037     \addtocontents{\@nameuse{ext@#2}}{%
8038         \protect\LWRsetnextfloat%
8039         {\arabic{LWR@latestautopage}}%
8040         {\arabic{LWR@thisautoid}}%
8041     }%
8042 }% not TOC
8043 \LWR@origaddcontentsline{#1}{#2}{#3}%
8044 }

```

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.

```

8045 \AtBeginDocument{
8046 \let\LWR@origcaptionof\captionof
8047
8048 \renewcommand*{\captionof}{%
8049 \LWR@stoppars
8050 \LWR@origcaptionof%
8051 }
8052 }

8053 \end{warpHTML}

```

71 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a `<div>` of class `toc`, `lof`, or `lot`.

A “`sidetoc`” is provided which prints a subset of the TOC on the side of each page other than the homepage.

The regular \TeX infrastructure is used for TOC, along with some patches to generate HTML output.

for HTML output: 8054 `\begin{warpHTML}`

71.1 Reading and printing the TOC

`\LWR@myshorttoc` $\{\langle toc/lof/lot/sidetoc \rangle\}$

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the @ character into a normal letter to allow formatting commands in the section names.

Unlike in regular \TeX , the file is not reset after being read, since the sideroc may be referred to again in each HTML page.

```
8055 \newcommand*\LWR@myshorttoc}[1]{%
8056 \LWR@traceinfo{\LWR@myshorttoc: #1}%
8057 \LWR@ensuredoingapar%
```

Only if the file exists:

```
8058 \IfFileExists{\jobname.#1}{%
8059 \LWR@traceinfo{\LWR@myshorttoc: loading}%
```



Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.

```
8060 \begingroup%
8061 \makeatletter%
```

Read in the TOC file:

```
8062 \@input{\jobname.#1}%
8063 \endgroup%
8064 }%
8065 }%
8066 \LWR@traceinfo{\LWR@myshorttoc: done}%
8067 }
```

`\LWR@subtableofcontents` $\{\langle toc/lof/lot \rangle\} \{\langle sectionstarname \rangle\}$

Places a TOC/LOF/LOT at the current position.

```
8068 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
8069 \@ifundefined{chapter}
8070 {\LWR@closeprevious{\LWR@depthsection}}
8071 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
8072 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
8073 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
8074 \LWR@htmlelementclass{nav}{#1}
```

Create the actual list:

```
8075 \LWR@myshorttoc{#1}
```

Close the nav:

```
8076 \LWR@htmlelementclassend{nav}{#1}
8077 }
```

```
\@starttoc {\langle ext\rangle}
```

Patch \@starttoc to encapsulate the TOC inside HTML tags:

```
8078 \let\LWR@orig@starttoc\@starttoc
8079
8080 \renewcommand{\@starttoc}[1]{
8081 \LWR@htmlelementclass{nav}{#1}
8082 \LWR@orig@starttoc{#1}
8083 \LWR@htmlelementclassend{nav}{#1}
8084 }
```

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.)

```
8085 \newbool{LWR@copiedsidetoc}
8086 \boolfalse{LWR@copiedsidetoc}
```

`\tableofcontents` Patch `\tableofcontents`, etc. to print footnotes first. **newfloat** uses `\listoffigures` for all future float types.

```
8087 \AtBeginDocument{
8088 \let\LWR@origtableofcontents\tableofcontents
8089
8090 \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:

```
8091 \ifboolexpr{bool{FormatWP} and bool{WPMarkTOC}}{
8092
8093 === table of contents ===
8094
8095 }
8096 {
```

Copy the `.toc` file to `.sidetoc` for printing the sideroc. The original `.toc` file is renewed when `\tableofcontents` is finished.

```
8097 \ifbool{LWR@copiedsidetoc}{}{%
8098 \LWR@copyfile{\jobname.toc}{\jobname.sidetoc}%
8099 \booltrue{LWR@copiedsidetoc}%
8100 }%
8101 \LWR@printpendingfootnotes
8102 \LWR@origtableofcontents
8103 }
8104 }% \tableofcontents
8105 }% AtBeginDocument
```

`\listoffigures`

```
8106 \let\LWR@origlistoffigures\listoffigures
8107
8108 \renewcommand*{\listoffigures}{
8109 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
8110
8111 === list of figures ===
8112
8113 }
8114 {
8115 \LWR@printpendingfootnotes
8116 \LWR@origlistoffigures
8117 }
8118 }
```

`\listoftables`

```

8119 \let\LWR@origlistoftables\listoftables
8120
8121 \renewcommand*{\listoftables}{
8122 \ifboolexpr{bool{FormatWP} and bool{WPMarkLOFT}}{
8123
8124 === list of tables ===
8125
8126 }
8127 {
8128     \LWR@printpendingfootnotes
8129     \LWR@origlistoftables
8130 }
8131 }

```

71.2 High-level TOC commands

`\listof` $\langle type \rangle$ $\langle title \rangle$

Emulate the `\listof` command from the **float** package (section 185). Used to create lists of custom float types. Also used to redefine the standard \TeX `\listoffigures` and `\listoftables` commands.

```

8132 \NewDocumentCommand{\listof}{m +m}{%
8133 \@ifundefined{l@#1}{%
8134     \csdef{l@#1}[2]{\hypertocfloat{1}{#1}{\csuse{ext@#1}}{##1}{##2}}%
8135 }{}%
8136 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
8137 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
8138 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
8139     \jobname.\csuse{ext@#1}\relax
8140 }

```

71.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sidetocname`, and may contain paragraphs.

css may be used to format the sideTOC:

CSS related to sideTOC:

nav.sidetoc: The entire sideroc.

div.sidetoctitle: The title.

div.sidetoccontents: The table of contents.

```
8141 \end{warpHTML}
```

for HTML & PRINT: 8142 \begin{warpall}

Ctrl SideTOCDepth Controls how deep the side-TOC gets. Use a standard \LaTeX section level similar to tocdepth.

```
8143 \newcounter{SideTOCDepth}
```

```
8144 \setcounter{SideTOCDepth}{1}
```

\sidetocname Holds the default name for the sideroc.

```
8145 \newcommand{\sidetocname}{Contents}
```

```
8146 \end{warpall}
```

for HTML output: 8147 \begin{warpHTML}

\LWR@sidetoc Creates the actual side-TOC.

```
8148 \newcommand*{\LWR@sidetoc}{
```

```
8149 \LWR@forcenewpage
```

```
8150 \LWR@stoppars
```

```
8151
```

The entire sideroc is placed into a nav of class sidetoc.

```
8152 \LWR@htmlclass{nav}{sidetoc}
```

```
8153
```

```
8154 \setcounter{tocdepth}{\value{SideTOCDepth}}
```

```
8155
```

The title is placed into a <div> of class sidetoctitle, and may contain paragraphs.

```
8156 \begin{BlockClass}{sidetoctitle}
```

```
8157 \sidetocname
```

```
8158 \end{BlockClass}
```

The table of contents is placed into a <div> of class `sidetoccontents`.

```
8159 \begin{BlockClass}{sidetoccontents}
8160 \LinkHome
8161
8162 \LWR@myshorttoc{sidetoc}
8163 \end{BlockClass}
8164 \LWR@html-elementclassend{nav}{sidetoc}
8165 }
```

71.4 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:

```
8166 \newcommand*{\LWR@numberline}[1]{%
8167 \LWR@sectionnumber{#1}\quad%
8168 }
8169
8170 \LetLtxMacro\numberline\LWR@numberline
```

`\hypertoc` $\{\langle 1: depth \rangle\} \{\langle 2: type \rangle\} \{\langle 3: name \rangle\} \{\langle 4: page \rangle\}$

Called by `\l@section`, etc. to create a hyperlink to a section.

The autopage label is always created just after the section opens.

#1 is depth

#2 is section, subsection, etc.

#3 the text of the caption

#4 page number

```
8171 \NewDocumentCommand{\hypertoc}{m m +m m}{%
8172 \LWR@traceinfo{hypertoc !#1!#2!#3!#4!}%
```

Respond to `tocdepth`:

```
8173 \ifthenelse{\cnttest{#1}{<=}}{\value{tocdepth}}{%
8174 \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```

8175 \LWR@subhyperrefclass{%
8176 \LWR@htmlrefsectionfilename{autopage-#4}\LWR@origpound\LWR@print@mbbox{autosec-#4}%
8177 }{#3}{toc#2}%
8178 \LWR@stoppars%
8179 }%
8180 {}%
8181 \LWR@traceinfo{hypertoc done}%
8182 }

```

Ctr lofdepth TOC depth for figures.

```

8183 \@ifclassloaded{memoir}{%{
8184 \newcounter{lofdepth}
8185 \setcounter{lofdepth}{1}
8186 }

```

Ctr lotdepth TOC depth for tables.

```

8187 \@ifclassloaded{memoir}{%{
8188 \newcounter{lotdepth}
8189 \setcounter{lotdepth}{1}
8190 }

```

\hypertocfloat {<1: depth>} {<2: type>} {<3: ext of parent>} {<4: caption>} {<5: page>}

#1 is depth

#2 is figure, table, etc.

#3 is lof, lot, of the parent.

#4 the text of the caption

#5 page number

```

8191 \newcommand{\hypertocfloat}[5]{%
8192 \LWR@startpars

```

If some float-creation package has not yet defined the float type's lofdepth counter, etc, define it here:

```

8193 \@ifundefined{c@#3depth}{%
8194 \newcounter{#3depth}%
8195 \setcounter{#3depth}{1}%
8196 }{}%

```

Respond to lofdepth, etc.:

```
8197 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
8198 \ifthenelse{\cnttest{#1}{<=}}{\arabic{#3depth}}{%
8199     \LWR@startpars%
```

Create an HTML link to filename#autoid-(float number), with text of the caption, of the given HTML class.

```
8200     \LWR@subhyperrefclass{%
8201     \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
8202     \LWR@origpound\LWR@print@mbox{autoid-\arabic{LWR@nextautoid}}}%
8203     {#4}{toc#2}%
8204     \LWR@stoppars%
8205 }{}%
8206 }
```

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.

```
8207 \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.

```
8208 \DeclareDocumentCommand{\l@chapter}{m m}
8209     {\hypertoc{0}{chapter}{#1}{#2}}
```

`\l@section` `{\langle name \rangle}{\langle page \rangle}`

```
8210 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
```

`\l@subsection` `{\langle name \rangle}{\langle page \rangle}`

```
8211 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
```

`\l@subsubsection` `{\langle name \rangle}{\langle page \rangle}`

```
8212 \renewcommand{\l@subsubsection}[2]{\hypertoc{3}{subsubsection}{#1}{#2}}
```

```

\l@paragraph  {\langle name\rangle} {\langle page\rangle}

8213 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}

\l@subparagraph  {\langle name\rangle} {\langle page\rangle}

8214 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}

\l@figure  {\langle name\rangle} {\langle page\rangle}

8215 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}

\l@table  {\langle name\rangle} {\langle page\rangle}

8216 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

8217 \end{warpHTML}

```

72 Index and glossary

See:

[http://tex.stackexchange.com/questions/187038/
how-to-mention-section-number-in-index-created-by-imakeidx](http://tex.stackexchange.com/questions/187038/how-to-mention-section-number-in-index-created-by-imakeidx)

Index links are tracked by the counter LWR@autoindex. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

```

for HTML output: 8218 \begin{warpHTML}

8219 \newcounter{LWR@autoindex}
8220 \setcounter{LWR@autoindex}{0}
8221
8222 \newcounter{LWR@autoglossary}
8223 \setcounter{LWR@autoglossary}{0}

\printindex

8224 \let\LWR@origprintindex\printindex
8225
8226 \renewcommand*\printindex

```

```

8227 {
8228 \LWR@startpars
8229 \LWR@origprintindex
8230 }

```

Env theindex

```

8231 \@ifundefined{chapter}
8232 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}
8233 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}
8234
8235 \renewenvironment*{theindex}{%
8236 \LWR@indexsection\indexname}%
8237 \let\item\LWR@indexitem%
8238 \let\subitem\LWR@indexsubitem%
8239 \let\subsubitem\LWR@indexsubsubitem%
8240 }{}

```

\LWR@indexitem

```

8241 \newcommand*\LWR@indexitem>{
8242
8243 \InlineClass{indexitem}{}
8244 }

```

\LWR@indexitem

```

8245 \newcommand*\LWR@indexsubitem>{
8246
8247 \InlineClass{indexsubitem}{}
8248 }

```

\LWR@indexitem

```

8249 \newcommand*\LWR@indexsubsubitem>{
8250
8251 \InlineClass{indexsubsubitem}{}
8252 }

```

\@wrindex {<term>} Redefined to write the LWR@autoindex counter instead of page

```

8253 \def\LWR@wrindex#1{%
8254 \addtocounter{LWR@autoindex}{1}%
8255 \LWR@newlabel{LWRindex-\arabic{LWR@autoindex}}%
8256 \protected@write\@indexfile{%
8257 {\string\indexentry{#1}{\arabic{LWR@autoindex}}}%

```

```

8258 \endgroup
8259 \@esphack}
8260
8261 \let\@wrindex\LWR@wrindex

```

`\@wrglossary {<term>}` Redefined to write the LWR@latestautopage counter instead of page

```

8262 \def\@wrglossary#1{%
8263 \addtocounter{LWR@autoglossary}{1}%
8264 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%
8265 \protected@write\@glossaryfile{%
8266 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%
8267 \endgroup
8268 \@esphack}

```

`\hyperindexref {<autosecnumber>}`

`\hyperindexref{web address}` is inserted into *.ind by the xindy style file lwarp.xdy

```

8269 \newcommand*\hyperindexref[1]{\nameref{LWRindex-#1}}

8270 \end{warpHTML}

```

for PRINT output: A null command for print mode, in case **hyperref** was not used:

```

8271 \begin{warpprint}
8272 \newcommand*\hyperindexref[1]{#1}
8273 \end{warpprint}

```

for HTML & PRINT: For the **glossaries** package, try to prevent an error where `\glo@name` was not found:

```

8274 \begin{warppall}
8275 \providecommand*\glo@name{}
8276 \end{warppall}

```

73 Bibliography presentation

for HTML output: 8277 \begin{warpHTML}

`\bibliography {<filenames>}`

Modified to use the base jobname instead of the `_html` jobname.

```

8278 \def\bibliography#1{%
8279     \if@filesw
8280     \immediate\write\@auxout{\string\bibdata{#1}}%
8281     \fi
8282 %     \@input@{\jobname.bbl}% original
8283     \begingroup%
8284     \@input@{\BaseJobname.bbl}% lwarp
8285     \endgroup%
8286 }

```

`\@biblabel` `{\text-refnumber}`

```

8287 \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>

```

8288 \AtBeginDocument{
8289 \AtBeginEnvironment{thebibliography}{
8290 \providecommand*\LWR@newem}[1]{\textit{#1}}
8291
8292 \renewrobustcmd{\em}{%
8293     \begingroup
8294     \gdef\LWR@em@after{\LWR@em@finish\LWR@newem}%
8295     \afterassignment\LWR@em@after
8296     \toks@\bgroup
8297 }
8298
8299 \def\LWR@em@finish#1{%
8300     \xdef\LWR@em@after{\noexpand#1{\the\toks@}}%
8301     \endgroup
8302     \LWR@em@after\egroup
8303 }
8304 }% \AtBeginEnvironment{thebibliography}
8305 }% \AtBeginDocument

8306 \end{warpHTML}

```

74 Restoring original formatting

`\LWR@restoreorigformatting` Used to temporarily restore the print-mode meaning of a number of formatting, graphics, and symbols-related macros while generating SVG math or a lateximage.

Must be used inside a group.

Sets `\LWR@formatting` to print until the end of the group.

A number of packages will `\appto` additional actions to this macro.

Various packages add to this macro using `\appto`.

```
for HTML output: 8307 \begin{warpHTML}

8308 \newcommand*{\LWR@restoreorigformatting}{%
8309 \LWR@traceinfo{\LWR@restoreorigformatting}%
8310 \renewcommand*{\LWR@formatting}{print}%
8311 \linespread{1}%

8312 \LetLtxMacro\caption@begin\LWR@origcaption@begin%
8313 \LetLtxMacro\caption@end\LWR@origcaption@end%
8314 \let\par\LWR@origpar%

8315 \LetLtxMacro\ref\LWR@origref%    {} syntax highlighting

8316 \let\normalsize\LWR@orignormalsize%
8317 \let\small\LWR@origsmall%
8318 \let\footnotesize\LWR@origfootnotesize%
8319 \let\scriptsize\LWR@origscriptsize%
8320 \let\tiny\LWR@origtiny%
8321 \let\large\LWR@origlarge%
8322 \let\Large\LWR@origLarge%
8323 \let\LARGE\LWR@origLARGE%
8324 \let\huge\LWR@orighuge%
8325 \let\Huge\LWR@origHuge%

8326 \LWR@select@print@hspace%

8327 \LetLtxMacro\hfill\LWR@orighfill%
8328 \LetLtxMacro\hfil\LWR@orighfil%
8329 \LetLtxMacro\hrulefill\LWR@orighrulefill%
8330 \LetLtxMacro\dotfill\LWR@origdotfill%
8331 \let\hss\LWR@orighss%
8332 \let\llap\LWR@origllap%
8333 \let\rlap\LWR@origrlap%
8334 \let\hfilneg\LWR@orighfilneg%
```

```
8335 \let\raggedright\LWR@origraggedright%
8336 \let\raggedleft\LWR@origraggedleft%
8337 \let\centering\LWR@origcentering%

8338 \let\,\LWR@origcomma% disable HTML short unbreakable space
8339 \let\thinspace\LWR@origthinspace% disable HTML short unbreakable space
8340 \let\textellipsis\LWR@origtextellipsis%
8341 \let\textless\LWR@origtextless%
8342 \let\textgreater\LWR@origtextgreater%
8343 \LetLtxMacro\textrm\LWR@origtextrm%
8344 \LetLtxMacro\textsf\LWR@origtextsf%
8345 \LetLtxMacro\texttt\LWR@origtexttt%
8346 \LetLtxMacro\textbf\LWR@origtextbf%
8347 \LetLtxMacro\textmd\LWR@origtextmd%
8348 \LetLtxMacro\textit\LWR@origtextit%
8349 \LetLtxMacro\textsl\LWR@origtextsl%
8350 \LetLtxMacro\textsc\LWR@origtextsc%
8351 \LetLtxMacro\textup\LWR@origtextup%
8352 \LetLtxMacro\textnormal\LWR@origtextnormal%
8353 \LetLtxMacro\emph\LWR@origemph%
8354 \LetLtxMacro\rmfamily\LWR@origrmfamily%
8355 \LetLtxMacro\sffamily\LWR@origsffamily%
8356 \LetLtxMacro\ttfamily\LWR@origttfamily%
8357 \LetLtxMacro\bfseries\LWR@origbfseries%
8358 \LetLtxMacro\mdseries\LWR@origmdseries%
8359 \LetLtxMacro\upshape\LWR@origupshape%
8360 \LetLtxMacro\slshape\LWR@origslshape%
8361 \LetLtxMacro\scshape\LWR@origscshape%
8362 \LetLtxMacro\itshape\LWR@origitshape%
8363 \LetLtxMacro\em\LWR@origem%
8364 \LetLtxMacro\normalfont\LWR@orignormalfont%
8365 \let\sp\LWR@origsp%
8366 \let\sb\LWR@origsb%
8367 \LetLtxMacro\textsuperscript\LWR@origtextsuperscript%
8368 \LetLtxMacro\@textsuperscript\LWR@orig@textsuperscript%
8369 \LetLtxMacro\textsubscript\LWR@origtextsubscript%
8370 \LetLtxMacro\@textsubscript\LWR@orig@textsubscript%
8371 \LetLtxMacro\underline\LWR@origunderline%
8372 \let~\LWR@origtilde%
8373 \let\enskip\LWR@origenskip%
8374 \let\quad\LWR@origquad%
8375 \let\qquad\LWR@origqquad%
8376 \LetLtxMacro\tabular\LWR@origtabular%
8377 \LetLtxMacro\endtabular\LWR@origendtabular%
8378 \LetLtxMacro\noalign\LWR@orignoalign%
8379 \LetLtxMacro\hline\LWR@orighline%
8380 \let\newline\LWR@orignewline%
8381 \LetLtxMacro\includegraphics\LWR@origincludegraphics%
8382 \let\TeX\LWR@origTeX%
```

```

8383 \let\LaTeX\LWR@origLaTeX%
8384 \let\LaTeXe\LWR@origLaTeXe%
8385 \renewcommand*{\Xe}{X\textsubscript{E}}%

8386 \LetLtxMacro\@ensuredmath\LWR@origensuredmath%
8387 %
8388 \LWR@restoreorigaccents%
8389 \LWR@restoreoriglists%
8390 %
8391 \LWR@Fbcancel%
8392 }

8393 \end{warpHTML}

```

75 Math

75.1 Limitations

75.1.1 Rendering tradeoffs

- | | |
|-----------------------|--|
| Math rendering | Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine. |
| SVG files | Rendering math as images creates a new svg file for each expression, except that an MD5 hash is used to combine identical duplicates of the same inline math expression into a single file, which must be converted to svg only once. Display math is still handled as individual files, since it may contain labels or references which are likely to change. |
| SVG inline | The svg images are currently stored separately, but they could be encoded in-line directly into the HTML document. This may reduce the number of files and potentially speed loading the images, but slows the display of the rest of the document before the images are loaded. |
| PNG files | Others \TeX -to-HTML converters have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but svg files are the preferred approach for scalable graphics. |
| MathML | Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than svg drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 10 regarding EPUB output with MATHJAX. |

75.1.2 SVG option

SVG math option For svg math, math is rendered as usual by \LaTeX into the initial PDF file using the current font¹⁷, then is captured from the PDF and converted to SVG graphics via a number of utility programs. The SVG format is a scalable-vector web format, so math may be typeset by \LaTeX with its fine control and precision, then displayed or printed at any size, depending on (sometimes broken) browser support. An HTML `alt` attribute carries the \LaTeX code which generated the math, allowing copy/paste of the \LaTeX math expression into other documents.


SVG image font size For the `lateximage` environment, the size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, which defaults to:

```
\renewcommand{\LateximageFontSizeName}{normalsize}
```

For inline SVG math, font size is instead controlled by `\LateximageFontScale`, which defaults to:

```
\newcommand*{\LateximageFontScale}{.75}
```

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

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

75.1.3 MATHJAX option

MATHJAX math option 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.

¹⁷See section 372 regarding fonts and fractions.

75.1.4 Customizing MATHJAX

MATHJAX does not have preexisting support every possible math function. Additional MATHJAX function definitions may be defined. These will be declared at the start of each HTML page, and thus will have a global effect.

Examples:

```
\CustomizeMathJax{
  \newcommand{\expval}[1]{\langle#1\rangle}
  \newcommand{\abs}[1]{\lvert#1\rvert}
}
\CustomizeMathJax{\newcommand{\arsinh}{\text{arsinh}}}
\CustomizeMathJax{\newcommand{\arcosh}{\text{arcosh}}}
\CustomizeMathJax{\newcommand{\NN}{\mathbb{N}}}
```

75.1.5 MATHJAX limitations

MATHJAX limitations

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 9.6.12.

⚠ siunitx inside an equation

tabbing

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

⚠ other macros and packages

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

75.1.6 Catcode changes

preamble macros with math

The math shift character `$` is not set for HTML output until after the preamble. Macros defined in the preamble which contain `$` must be enclosed between `\StartDefiningMath` and `\StopDefiningMath` to temporarily change to the HTML meaning of `$`:

```
\StartDefiningMath
\newcommand{...}
\StopDefiningMath
```

As an alternative, use `\(` and `\)` instead of `$`, in which case `\StartDefiningMath` and `\StopDefiningMath` are not necessary.

If a package defines macros using `$`, it may be necessary to use `\StartDefiningMath` and `\StopDefiningMath` before and after loading the package.

75.1.7 Dynamic math

inline dynamic math

An inline math expression is usually converted to a reusable hashed svg math image, or a MathJax expression. The hash or expression depends on the contents of the math expression. In most cases this math expression is static, such as $x+1$, so the image can be reused for multiples instances of the same expression. In some cases, the math expression includes a counter or other object which may change between uses. The macro `\StartDynamicMath` may be used before a dynamic math expression, and `\StopDynamicMath` after. Doing so tells **lwarp** to use an unhashed svg math image, even if MathJax is in use. See section 41.

75.1.8 Display math

`\displaymathnormal`

By default, or when selecting `\displaymathnormal`, math display environments print their contents in MATHJAX, and render their contents in svg math as well as use their contents in the `alt` tag of HTML output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated Tikz pictures, compilation will fail.

`\displaymathother`

When selecting `\displaymathother`, it is assumed that the contents are more complicated than “pure” math. An example is an elaborate Tikz picture, which will not

render in MATHJAX and will not make sense as an HTML alt tag. In this mode, MATHJAX is turned off, math display environments become SVG images, even for MATHJAX, and the HTML alt tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as Tikz pictures are more likely to compile successfully.

75.2 HTML alt tag names

Redefinable names for the HTML alt tags, for translation according to the reader's native language.

for HTML & PRINT: 8394 `\begin{warpall}`

`\mathimagenam` The HTML alt tag for an SVG math image.
 Default: "math image"

8395 `\newcommand*{\mathimagenam}{math image}`

`\packagediagramname` Appended to the lateximage HTML alt tag for the images generated by many packages.
 Default: "diagram"

8396 `\newcommand*{\packagediagramname}{diagram}`

8397 `\end{warpall}`

75.3 Inline and display math

for HTML output: 8398 `\begin{warpHTML}`

Ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

8399 `\newcounter{LWR@externalfilecnt}`

Bool `LWR@indisplaymathimage` True if processing display math for SVG output. Inside a lateximage, display math is only set to print-mode output if LWR@indisplaymathimage is false. Used to avoid nullifying display math before it has been completed.

8400 `\newbool{LWR@indisplaymathimage}`

`\$` Plain dollar signs appearing in the HTML output may be interpreted by MATHJAX to be math shifts. For a plain text dollar `\$`, use an HTML entity to avoid it being

interpreted by MATHJAX, unless are inside a lateximage, in which case it will not be seen by MATHJAX.

```

8401 \let\LWR@origtextdollar\$
8402
8403 \renewcommand*{\$}{\%
8404 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
8405 {\LWR@origtextdollar}%
8406 {\HTMLentity{dollar}}}%
8407 }

```

File `lwarp_baseline_marker.png` A marker to be used to help **pdfcrop** identify the inline math baseline and width. If either **graphicx** or **graphics** is loaded, this marker is placed at the lower left and lower right corners of the inline math. **pdfcrop** is then able to identify the width of the image, and also the height of an image such as a horizontal dash which does not otherwise touch the baseline.

A marker with alpha or opacity of 0% is not registered by **pdfcrop**, so the marker is a small square block of 1% alpha, which seems to work while still being effectively invisible in the final SVG image.

If **graphicx** is loaded, this marker is sized as a tiny 1 sp square. If **graphics** is loaded, this marker is used at its default size of around .25 pt. If neither graphics package is loaded, the marker is replaced by a 10 sp horizontal space, and there is no assistance for determining baseline or width of the inline math image. The best results are obtained when using **graphicx**.

`\LWR@addbaselinemarker` Places a small marker in an svg inline image. If **graphics** or **graphicx** are loaded, the marker is a mostly transparent image. If neither is loaded, no marker is used.

```

8408 \AtBeginDocument{
8409
8410 \IfFileExists{lwarp_baseline_marker.png}%
8411 {
8412   \@ifpackageloaded{graphicx}{
8413     \newcommand*{\LWR@addbaselinemarker}{\%
8414       \LWR@originincludegraphics[%
8415         width=10sp,height=10sp%
8416       ]{lwarp_baseline_marker.png}%
8417   }
8418 }{
8419   \@ifpackageloaded{graphics}{
8420     \newcommand*{\LWR@addbaselinemarker}{\%
8421       \LWR@originincludegraphics{lwarp_baseline_marker.png}%
8422     }
8423   }{
8424     \PackageWarning{lwarp}{Load graphicx or graphics
8425       for improved SVG math baselines,}

```

```

8426         \newcommand*{\LWR@addbaselinemarker}{%
8427             \hspace*{10sp}%
8428         }
8429     }
8430 }
8431 }{% lwarp_baseline_marker.png not present
8432     \PackageWarning{lwarp}{File lwarp_baseline_marker.png is not installed alongside
8433         the lwarp-*.sty files, so SVG math baselines may not be accurate,}
8434     \newcommand*{\LWR@addbaselinemarker}{%
8435         \hspace*{10sp}%
8436     }
8437 }
8438
8439 }% AtBeginDocument

```

`\LWR@subsingledollar` * $\langle 2: \textit{alt text} \rangle \langle 3: \textit{add'l hashing} \rangle \langle 4: \textit{math expression} \rangle$

For inline math. Uses MathJax, or for svg math the image is measured and adjusted to the baseline of the HTML output, and placed inside a `lateximage`.

image filename hashing If starred, a hashed filename is used. If so, the hash is based on the `alt` tag and also the additional hashing argument.

This may be used to provide an expression with a simple `alt` tag but also enough additional information to provide a unique hash.

An example is when the expression is a complicated \TeX expression, which would not copy/paste well. A simplified tag may be used, while the complicated expression is duplicated in the additional hashing argument.

Another example is when the expression is simple, but the image depends on options. These options may be decoded into text form and included in the additional hashing argument in order to make the hash unique according to the set of options, even if the simple `alt` tag is still the same.

```

8440 \newlength{\LWR@singledollarwidth}
8441 \newlength{\LWR@singledollarheight}
8442 \newlength{\LWR@singledollardepth}
8443
8444 \newsavebox{\LWR@singledollarbox}
8445
8446 \NewDocumentCommand{\LWR@subsingledollar}{s m m m}{%
8447     \LWR@traceinfo{\LWR@subsingledollar}%

8448     \ifnumcomp{\value{\LWR@lateximagedepth}}{>}{0}%
8449     {%
8450     \LWR@traceinfo{\LWR@subsingledollar: already in a lateximage}%
8451     #4% contents

```

```

8452 }%
8453 {% not in a lateximage
8454 \begingroup%

```

MathJax cannot parse the often complicated \TeX expressions which appear in the various uses of `\ensuredmath`. `\ensuremath` forces the `alt` tag to “(math image)”, as translated according to `\mathimagername`. If this is the case, force the use of a `lateximage` even if MathJax. Likewise for `siunitx` if `parse-numbers=false`.

If MathJax, or if formatting math for a word processor, and not `\ensuredmath`, and not a dynamic math expression, print the math expression:

```

8455 \ifboolexpr{%
8456   (
8457     bool{mathjax} or
8458     ( bool{FormatWP} and bool{WPMarkMath} )
8459   ) and
8460   ( not test { \ifstrequal {#2} {(\mathimagername)} } ) and % from \ensuredmath
8461   ( not bool{LWR@dynamicmath} )
8462 }%

```

For MATHJAX, print the math between `\(` and `\)`:

```

8463 {%
8464   \LWR@traceinfo{LWR@subsingledollar: Mathjax}%
8465   {\textbackslash\LWR@HTMLsanitize{#4}\textbackslash}%
8466 }% mathjax

```

For SVG, print the math inside a `lateximage`, with an `<alt>` tag of the \TeX code, and a CSS style to control the baseline adjustment.

```

8467 {% not mathjax
8468 \LWR@traceinfo{LWR@subsingledollar: NOT mathjax, or is ensuremath, or is dynamic}%

```

Measure the depth, width, and height of the math image:

```

8469   \begingroup%

```

Temporarily disable formatting while measuring the image parameters:

```

8470   \LWR@restoreorigformatting%
8471   \RenewDocumentEnvironment{lateximage}{s o o o}{-}{-}% inside group
8472   \LWR@orignormalsize%

```

Temporarily set font for the HTML PDF output:

```

8473   \LWR@traceinfo{Using font family \LWR@f@family}%
8474   \csuse{LWR@orig\LWR@f@family family}%

```

```

8475 \LWR@traceinfo{Using font series \LWR@f@series}%
8476 \csuse{LWR@orig\LWR@f@series series}%
8477 \LWR@traceinfo{Using font shape \LWR@f@shape}%
8478 \csuse{LWR@orig\LWR@f@shape shape}%

```

lateximagedepth must be nested to avoid generating paragraph tags. \mathcal{AMS} math modifies the `\text` macro such that `\addtocounter` does not always occur as expected. Lower-level code is used instead.

```

8479 \global\advance\c@LWR@lateximagedepth 1\relax%

```

Typeset and save the contents, depending on how they were generated:

SVG math: `\LWR@origensuredmath` is part of argument #4.

SVG math \ensuremath: `\LWR@origensuredmath` is part of argument #4.

SVG dynamic math: `\LWR@origensuredmath` is part of argument #4.

Mathjax: Argument #4 is the contents of the math expression without `\LWR@origensuredmath`. This case is handled above.

Mathjax \ensuremath: `\LWR@origensuredmath` is part of argument #4.

Mathjax dynamic math: Argument #4 is the contents of the math expression without `\LWR@origensuredmath`, so `\LWR@origensuredmath` is added below.

\ifmmode: Included “just in case”.

```

8480 \ifmmode%
8481 \global\sbox{\LWR@singledollarbox}{#4}%
8482 \else%
8483 \ifbool{LWR@dynamicmath}{%
8484 \ifbool{mathjax}{%
8485 \global\sbox{\LWR@singledollarbox}{\LWR@origensuredmath{#4}}%
8486 }{%
8487 \global\sbox{\LWR@singledollarbox}{#4}%
8488 }%
8489 }{%
8490 \global\sbox{\LWR@singledollarbox}{#4}%
8491 }%
8492 \fi%

```

Add a small and almost transparent marker at the depth of the image.

A math minus sign has the same depth as a plus, even though it does not draw anything below the baseline. This means that **pdfcrop** would crop the image without depth. The marker below the baseline is seen by **pdfcrop** and preserves the depth.

```

8493   \global\sbox{\LWR@singledollarbox}{%
8494       \usebox{\LWR@singledollarbox}%
8495       \hspace*{-10sp}%
8496       \raisebox{-\dp\LWR@singledollarbox}{%
8497           \LWR@addbaselinemarker%
8498       }%
8499   }%

```

More low-level code to undo the counter change.

```

8500   \global\advance\c@LWR@lateximagedepth -1\relax% Due to AmS \text macro.

```

Measure the depth:

```

8501   \setlength{\LWR@singledollardepth}{%
8502       \LateximageFontScale\dp\LWR@singledollarbox%
8503   }%

```

Make the length a global change:

```

8504   \global\LWR@singledollardepth=\LWR@singledollardepth%

```

Likewise for width:

```

8505   \setlength{\LWR@singledollarwidth}{%
8506       \LateximageFontScale\wd\LWR@singledollarbox%
8507   }%
8508   \global\LWR@singledollarwidth=\LWR@singledollarwidth%

```

Likewise for total height:

```

8509   \setlength{\LWR@singledollarheight}{%
8510       \LateximageFontScale\ht\LWR@singledollarbox%
8511   }%
8512   \addtolength{\LWR@singledollarheight}{%
8513       \LateximageFontScale\dp\LWR@singledollarbox%
8514   }%
8515   \global\LWR@singledollarheight=\LWR@singledollarheight%

8516   \endgroup%

```

Set a style for the the height or width. The em unit is used so that the math scales according to the user's selected font size.

Start with the greater of the width or the height, biased towards the width:

```

8517   \ifdimgreater{\LWR@singledollarwidth}{.7\LWR@singledollarheight}{%
8518       \def\LWR@singledollarstyle{%

```

```

8519         width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
8520     }%
8521 }{%
8522     \def\LWR@singledollarstyle{%
8523         height:\LWR@convertto{em}{\the\LWR@singledollarheight} em%
8524     }%
8525 }%

```

If a very narrow width, use the height.

```

8526     \ifdimless{\LWR@singledollarwidth}{.2em}%
8527     {%
8528         \def\LWR@singledollarstyle{%
8529             height:\LWR@convertto{em}{\the\LWR@singledollarheight} em%
8530         }%
8531     }%
8532     {}%

```

If very wide and short, use the width:

```

8533     \ifdimless{\LWR@singledollarheight}{.2em}%
8534     {%
8535         \def\LWR@singledollarstyle{%
8536             width:\LWR@convertto{em}{\the\LWR@singledollarwidth} em%
8537         }%
8538     }%
8539     {}%

```

If there is significant text depth, add the depth to the style.

```

8540     \ifdimgreater{\LWR@singledollardepth}{0.05ex}{%
8541         \def\LWR@singledollardepthstyle{%
8542             \ ; % extra space
8543             \LWR@print@mbbox{%
8544                 vertical-align:-\LWR@convertto{em}{\the\LWR@singledollardepth} em%
8545             } % extra space
8546         }%
8547     }{%
8548         \def\LWR@singledollardepthstyle{}%
8549     }%

```

Create the `lateximage` using the alternate tag and the computed size and depth. The star causes `lateximage` to use an MD5 hash as the filename. When hashing, also include the current font and color in the hash.

```

8550     \ifbool{LWR@dynamicmath}{%
8551         \LWR@traceinfo{subsingledollar: dynamic}%
8552         \begin{lateximage}% no hashing
8553             [(\mathimagenam)]% alt tag

```

```

8554          []% no add'l hashing
8555          [\LWR@singledollarstyle \LWR@singledollardepthstyle]% CSS
8556      }{%
8557          \LWR@traceinfo{subsingledollar: static}%
8558          \IfValueTF{#1}{%
8559              \LWR@findcurrenttextcolor% sets \LWR@tempcolor
8560              \begin{lateximage}*% use hashing
8561                  [#2]% alt
8562                  [% addl' hashing
8563                      #3%
8564                      FM\LWR@f@family%
8565                      SR\LWR@f@series%
8566                      SH\LWR@f@shape%
8567                      CL\LWR@tempcolor%
8568                  ]%
8569                  [\LWR@singledollarstyle \LWR@singledollardepthstyle]% CSS
8570          }{%
8571              \begin{lateximage}% no hashing
8572                  [#2]% alt
8573                  []% no add'l hashing
8574                  [\LWR@singledollarstyle \LWR@singledollardepthstyle]% CSS
8575          }%
8576      }%

```

Place small and almost transparent markers on the baseline at the left and right edges of the image. These markers are seen by **pdfcrop**, and force vertically-centered objects such as a dash to be raised off the baseline in the cropped image, and also force the total width and left/right margins to be correct. (Except that in some fonts a character may exceed the bounding box, and thus may appear wider than expected when converted to an image.)

```

8577      \LWR@addbaselinemarker%
8578      \hspace*{-10sp}%

```

Typeset the contents:

```

8579      \usebox{\LWR@singledollarbox}%

```

The closing baseline marker:

```

8580      \hspace*{-10sp}%
8581      \LWR@addbaselinemarker%

8582      \end{lateximage}%
8583      %
8584      }% not mathjax
8585      \endgroup%
8586      }% not in a lateximage

```

```
8587 \LWR@traceinfo{LWR@subsingledollar: done}%
8588 }
```

```
8589 \LetLtxMacro\LWR@origdollar$
8590 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting
```

```
8591 \LetLtxMacro\LWR@origopenparen\(  
8592 \LetLtxMacro\LWR@origcloseparen\  
8593 \LetLtxMacro\LWR@origopenbracket\  
8594 \LetLtxMacro\LWR@origclosebracket\]
```

\$ Redefine the dollar sign to place math inside a lateximage, or use MATHJAX:
\$\$

```
8595 \begingroup
8596 \catcode'\$=\active%
8597 \protected\gdef$\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

Used by **chemformula** to escape single-dollar math:

```
8598 \protected\gdef\LWR@newsingledollar{\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%

```

\LWR@doubledollar Redefine the double dollar sign to place math inside a lateximage, or use MATHJAX:

```
8599 \protected\gdef\LWR@doubledollar$#1$${%

```

If MATHJAX or formatting for a word processor, print the \LaTeX expression:

```
8600 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%

```

For MATHJAX, print the math between `\[` and `\]`:

```
8601 {
8602
8603   \textbackslash[%
8604   \LWR@HTMLsanitize{#1}%
8605   \textbackslash]
8606
8607 }% mathjax

```

For SVG, print the math inside a lateximage, with an `<alt>` tag of the \LaTeX code:

```
8608 {% not mathjax
8609   \begin{BlockClass}{displaymath}%
8610   \LWR@newautoidanchor%
8611   \booltrue{LWR@indisplaymathimage}%
8612   \begin{lateximage}%
8613   [%

```

```

8614      \textbackslash{[] % extra space
8615      \LWR@HTMLsanitize{#1} % extra space
8616      \textbackslash{[]}%
8617  ]%
8618  \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
8619  \end{lateximage}%
8620  \end{BlockClass}%
8621 }% not mathjax
8622 }%

```

`\LWR@singledollar` $\{\langle alt\ text\rangle\}\{\langle math\ expression\rangle\}$

```

8623 \protected\gdef\LWR@singledollar#1${%
8624 \ifbool{mathjax}{%
8625   \LWR@subsingledollar*%
8626   {% alt tag
8627     \textbackslash( %
8628     \LWR@HTMLsanitize{#1} % extra space
8629     \textbackslash)%
8630   }%
8631   {singledollar}% add'l hashing
8632   {#1}% contents
8633 }{% not mathjax
8634   \LWR@subsingledollar*%
8635   {% alt tag
8636     \textbackslash( %
8637     \LWR@HTMLsanitize{#1} % extra space
8638     \textbackslash)%
8639   }%
8640   {singledollar}% add'l hashing
8641   {\LWR@origensuredmath{#1}}}% contents
8642 }% not mathjax
8643 }

```

`\(` Redefine to the above dollar macros.

`\)`

```

8644 \protected\gdef\(#1\){$#1$}
8645 \protected\gdef\[#1\]{$$#1$$}
8646
8647 \endgroup
8648
8649 \LetLtxMacro\LWR@openbracketnormal\[
8650 \LetLtxMacro\LWR@closebracketnormal\]

```

`\@ensuredmath` $\{\langle expression\rangle\}$

If MathJax, a `lateximage` is used, since `\ensuremath` is often used for complex \TeX expressions which MathJax may not render. If `svg math`, a hashed file is used with a simple `alt` tag, but additional hashing provided by the contents.

```

8651 \LetLtxMacro\LWR@origensuredmath\@ensuredmath
8652
8653 \renewcommand{\@ensuredmath}[1]{%
8654 \ifbool{mathjax}{%
8655     \LWR@subsingledollar*{(\mathimagername)}{%
8656         \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}}%
8657     }{\relax%
8658         \LWR@origensuredmath{#1}}%
8659     }%
8660 }{% SVG math

```

If already inside a `lateximage` in `math` mode, continue as-is.

```

8661 \ifmmode%
8662     \LWR@origensuredmath{#1}%
8663 \else%

```

Create an inline `math lateximage` with a simple `alt` tag and additional hashing according to the contents.

```

8664     \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
8665     {\LWR@origensuredmath{#1}}%
8666     {%
8667         \LWR@subsingledollar*{(\mathimagername)}{%
8668             \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}}%
8669         }{%
8670             \LWR@origensuredmath{#1}}%
8671         }%
8672     }%
8673 \fi%
8674 }%
8675 }

```

Remove the old `math` and `displaymath` environments:

```

8676 \let\math\relax
8677 \let\endmath\relax
8678 \let\displaymath\relax
8679 \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`.

```

8680 \NewEnviron{math}{\expandafter\(\BODY\)}

```

Env LWR@displaymathnormal Set math mode then typeset the body of what was between the begin/end. See the **environ** package for \BODY.

```
8681 \NewEnviron{LWR@displaymathnormal}{\expandafter\[\BODY\]\@ignoretrue}
```

Set the default displaymath to the normal version:

```
8682 \LetLtxMacro\[\LWR@openbracketnormal%
8683 \LetLtxMacro\]\LWR@closebracketnormal%
8684 \LetLtxMacro\displaymath\LWR@displaymathnormal%
8685 \LetLtxMacro\enddisplaymath\endLWR@displaymathnormal%
```

Env LWR@displaymathother A version of displaymath which can handle complicated objects, but does not supply MATHJAX or HTML alt tags.

```
8686 \newenvironment{LWR@displaymathother}
8687 {%
8688   \begin{BlockClass}{displaymath}%
8689   \LWR@newautoidanchor%
8690   \booltrue{LWR@indisplaymathimage}%
8691   \begin{lateximage}%
8692   [(display math)]%
8693   \LWR@origdollar\LWR@origdollar%
8694 }
8695 {%
8696   \LWR@origdollar\LWR@origdollar%
8697   \end{lateximage}%
8698   \end{BlockClass}%
8699 }
```

Env LWR@equationother A version of displaymath which can handle complicated objects, but does not supply MATHJAX or HTML alt tags.

```
8700 \newenvironment{LWR@equationother}
8701 {%
8702   \begin{BlockClass}{displaymathnumbered}%
8703   \LWR@newautoidanchor%
8704   \booltrue{LWR@indisplaymathimage}%
8705   \begin{lateximage}%
8706   [(display math)]%
8707   \LWR@origequation%
8708 }
8709 {%
8710   \LWR@origendequation%
8711   \end{lateximage}%
8712   \end{BlockClass}%
8713 }
```

75.4 MATHJAX support

Ctrl LWR@nextequation Used to add one to compute the next equation number.

```
8714 \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 .

```
8715 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
8716 \ifcsdef{thechapter}{
8717 \InlineClass{hidden}{
8718 \textbackslash(
8719 \textbackslash{seteqsection \{\thechapter\}
8720 \textbackslash)}
8721 }
8722 }
8723 {}}% not using chapters
```

MATHJAX doesn't allow setting the equation number to 1:

```
8724 \ifthenelse{\cnttest{\value{equation}}>0}
8725 {
```

Tell MATHJAX that the next set of equations begins with the current \TeX equation number, plus one.

```
8726 \setcounter{LWR@nextequation}{\value{equation}}
8727 \addtocounter{LWR@nextequation}{1}
```

Place the MATHJAX command inside “\(" and “\)” characters, to be printed to HTML, not interpreted by \TeX .

```
8728 \InlineClass{hidden}{
8729 \textbackslash(
8730 \textbackslash{seteqnumber \{\arabic{LWR@nextequation}\}
8731 \textbackslash)}
8732 }
8733 {}}% not eq > 0
8734 }
```

```
\LWR@hidelatexequation {<environment>} {<contents>}
```

Creates the \LaTeX version of the equation inside an HTML comment.

```
8735 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```
8736 \LWR@stoppars
8737 \LWR@htmlopencomment
8738
```

Start the \LaTeX math environment inside the HTML comment:

```
8739 \begingroup
8740 \csuse{\LWR@orig#1}
```

While in the math environment, restore various commands to their \LaTeX meanings.

```
8741 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 75.7.1.

Print the contents of the equation:

```
8742 #2
```

End the \LaTeX math environment inside the HTML comment:

```
8743 \csuse{\LWR@origend#1}
8744 \endgroup
8745
```

Close the HTML comment and resume HTML paragraph handling:

```
8746 \LWR@htmlclosecomment
8747 \LWR@startpars
8748 }
```

```
\LWR@addmathjax {<environment>} {<contents>}
```

Given the name of a math environment and its contents, create a `MATHJAX` instance. The contents are printed to HTML output, not interpreted by \LaTeX .

```
8749 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the `MATHJAX` environment inside printed “\(" and “\)” characters.

```
8750 \LWR@origtilde\LWR@orignewline
8751 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
8752 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MATHJAX environment:

```
8753 \textbackslash{}end\{#1\}
8754 \LWR@orignewline
8755 }
```

75.5 Equation environment

Remember existing equation environment:

```
8756 \let\LWR@origequation\equation
8757 \let\LWR@origendequation\endequation
8758 \csletcs\LWR@origequation*{equation*}
8759 \csletcs\LWR@origendequation*{endequation*}
```

`\LWR@doequation` For SVG math output, the contents are typeset using the original equation inside a `lateximage`, along with an `<alt>` tag containing a detokenized copy of the \TeX source for the math.

For MATHJAX output, the contents are typeset in an original equation environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MATHJAX script.

```
8760 \newcommand*{\LWR@doequation}[2]{%
8761
```

If `mathjax` or `FormatWP`, print the \TeX expression:

```
8762 \ifbool{expr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }}{%
```

MATHJAX output:

```
8763 {
```

Print commands to synchronize MATHJAX's equation number and format to the current \TeX chapter/section and equation number:

```
8764 \LWR@synchronmathjax
```

Print the \LaTeX math inside an HTML comment:

```
8765 \LWR@hidelatexequation{#2}{#1}
8766 }
```

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`.

```
8767 {% not mathjax
```

Begin the `lateximage` with an `<alt>` tag containing the math source:

```
8768 \ifstrequal{#2}{equation*}{%
8769 \begin{BlockClass}{displaymath}%
8770 }{%
8771 \begin{BlockClass}{displaymathnumbered}%
8772 }%
8773 \LWR@newautoidanchor%
8774 \booltrue{LWR@indisplaymathimage}%
8775 \begin{lateximage}[%
8776 \ifstrequal{#2}{equation*}{%
8777 \ifdefequal{LWR@equationtag}{\theequation}{%
8778 % no tag was given
8779 }{%
8780 (\LWR@equationtag) % tag was given
8781 }%
8782 }{%
8783 (\LWR@equationtag) % automatic numbering
8784 }%
8785 \textbackslashbegin\{#2\}} % extra space
8786 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#1}} % extra space
8787 \textbackslashend\{#2\}}%
8788 ]% alt tag
```

Create the actual \LaTeX -formatted equation inside the `lateximage` using the contents of the environment.

```
8789 \csuse{LWR@orig#2}
8790 #1% contents collected by \collect@body
8791 \csuse{LWR@origend#2}
8792 \end{lateximage}%
8793 \end{BlockClass}
8794 }% not mathjax
8795 }
```

After the environment, if MATHJAX, print the math to the HTML output for MATHJAX processing:

```

8796 \newcommand*\LWR@doendequation}[1]{%
8797   \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8798   {%
8799     \LWR@addmathjax{#1}{\BODY}%
8800   }{}%
8801
8802 }
```

Remove existing equation environment:

```

8803 \let\equation\relax
8804 \let\endequation\relax
8805 \csletcs{equation*}{relax}
8806 \csletcs{endequation*}{relax}
```

Env `equation` The new equation environment is created with `\NewEnviron` (from the **environ** package), which stores the contents of its environment in a macro called `\BODY`.

```

8807 \NewEnviron{equation}
8808 {
8809   \LWR@doequation{\BODY}{equation}
8810 }
8811 [\LWR@doendequation{equation}]
8812
8813 \LetLtxMacro\LWR@equationnormal\equation
8814 \LetLtxMacro\LWR@endequationnormal\endequation
```

Env `equation*`

```

8815 \NewEnviron{equation*}
8816 {\LWR@doequation{\BODY}{equation*}}
8817 [\LWR@doendequation{equation*}]
8818
8819 \csletcs{LWR@equationnormalstar}{equation*}
8820 \csletcs{LWR@endequationnormalstar}{endequation*}
```

Remember the “less” version of equation, which use MATHJAX and alt tags, but does not support complicated contents such as some Tikz expressions.

```

8821 \LetLtxMacro\LWR@equationless\equation
8822 \LetLtxMacro\LWR@endequationless\endequation
8823 \csletcs{LWR@equationlessstar}{equation*}
8824 \csletcs{LWR@endequationlessstar}{endequation*}
```

75.6 `\displaymathnormal` and `\displaymathother`

`\displaymathnormal` By default, or when selecting `\displaymathnormal`, math display environments print their contents in MATHJAX, and render their contents in svg math as well as use their contents in the alt tag of HTML output. To do so, the contents are loaded into a macro for reuse. In some cases, such as complicated Tikz pictures, compilation will fail.

`\displaymathother` When selecting `\displaymathother`, it is assumed that the contents are more complicated than “pure” math. An example is an elaborate Tikz picture, which will not render in MATHJAX and will not make sense as an HTML alt tag. In this mode, MATHJAX is turned off, math display environments become svg images, even for MATHJAX, and the HTML alt tags become simple messages. The contents are internally processed as an environment instead of a macro argument, so complicated objects such as Tikz pictures are more likely to compile successfully.

`\displaymathnormal`
[simple math objects](#) Use when display math environments have simple math which is to sent to MATHJAX or included in HTML alt tags.

```
8825 \newcommand*{\displaymathnormal}{%
8826 \ifbool{LWR@origmathjax}{\booltrue{mathjax}}{\boolfalse{mathjax}}%
8827 \LetLtxMacro\[\LWR@openbracketnormal%
8828 \LetLtxMacro\]\LWR@closebracketnormal%
8829 \LetLtxMacro\displaymath\LWR@displaymathnormal%
8830 \LetLtxMacro\enddisplaymath\endLWR@displaymathnormal%
8831 \LetLtxMacro\equation\LWR@equationnormal%
8832 \LetLtxMacro\endequation\LWR@endequationnormal%
8833 \csletcs{equation*}{LWR@equationnormalstar}%
8834 \csletcs{endequation*}{LWR@endequationnormalstar}%
8835 }
```

`\displaymathother`
[complicated math objects](#) Use when display math environments have complicated objects which will not work with MathJax or should not be included in HTML alt tags. Complicated contents are more likely to compile correctly.

```
8836 \newcommand*{\displaymathother}{%
8837 \boolfalse{mathjax}%
8838 \LetLtxMacro\displaymath\LWR@displaymathother%
8839 \LetLtxMacro\enddisplaymath\endLWR@displaymathother%
8840 \LetLtxMacro\[\LWR@displaymathother%
8841 \LetLtxMacro\]\endLWR@displaymathother%
8842 \LetLtxMacro\equation\LWR@equationother%
8843 \LetLtxMacro\endequation\endLWR@equationother%
8844 \csletcs{equation*}{\displaymath}%
8845 \csletcs{endequation*}{\enddisplaymath}%
8846 }
```

```
8847 \end{warpHTML}
```

for PRINT output: 8848 \begin{warpprint}

Print-mode versions:

```
8849 \newcommand*{\displaymathnormal}{}
8850 \newcommand*{\displaymathother}{}
8851 \end{warpprint}
```

for HTML output: 8852 \begin{warpHTML}

75.7 AMS Math environments

75.7.1 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.

```
8853 \newbool{LWR@amsmultline}
8854 \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**.

```
8855 \newcommand*{\LWR@htmlmathlabel}[1]{%
8856 \LWR@traceinfo{LWR@htmlmathlabelb #1}%
```

If mathjax or FormatWP, print the \TeX expression:

```
8857 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8858 {%
```

The combined \TeX & HTML label is printed in a \text field:

```
8859 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
8860 \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.)

```
8861 \LWR@htmlclosecomment%
8862 \LWR@origltx@label{#1}%
8863 \LWR@htmlopencomment%
8864 }% text
8865 }% mathjax
8866 {%
8867 \LWR@origltx@label{#1}%
8868 }%
8869 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
8870 \newcommand*{\LWR@beginhideamsmath}{
8871 \LWR@stoppars
8872 \LWR@origtilde\LWR@orignewline
8873 \LWR@htmlopencomment
8874
8875 \begingroup
8876 \LWR@restoreorigformatting
8877 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```
8878 \newcommand*{\LWR@endhideamsmath}{
8879 \endgroup
8880
8881 \LWR@htmlclosecomment
8882 \LWR@orignewline
8883 \LWR@startpars
8884 }
```

75.7.2 Environment patches

The following **amsmath** environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an $\mathcal{A}\mathcal{M}\mathcal{S}$ package, and thus requires special handling.

For `svg math`: Each environment is encapsulated inside a `lateximage` environment, along with a special optional argument of `\LWR@amsmathbody` or `\LWR@amsmathbodynumbered` telling `lateximage` to use as the HTML `<alt>` tag the environment's contents which were automatically captured by the \mathcal{AMS} environment.

For `MATHJAX`: Each environment is syched with \TeX 's equation numbers, typeset with \TeX inside an HTML comment, then printed to HTML output for `MATHJAX` to process.

Env `eqnarray` This environment is not an \mathcal{AMS} environment and thus its body is not automatically captured, so the `environ` package is used to capture the environment into `\BODY`.

```
8885 \let\LWR@origeqnarray\eqnarray
8886 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
8887 \newbool{LWR@numbereqnarray}
8888 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from **fancybox**):

```
8889 \newcommand{\LWR@eqnarrayfactor}{%
```

If `mathjax` or `FormatWP`, print the \TeX expression:

```
8890 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8891 {%
```

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:

```
8892 \LWR@syncmathjax
8893 \boolfalse{LWR@amsmultline}
8894 \ifbool{LWR@numbereqnarray}
8895 {
```

If numbering the equations, execute a copy inside an HTML comment block:

```
8896 \LWR@beginhideamsmath
8897 \LWR@origeqnarray
8898 \BODY
8899 \LWR@origendeqnarray
8900 \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MATHJAX to interpret:

```
8901      \LWR@addmathjax{eqnarray*}{\BODY}
8902    }%
8903    {% not LWR@numbreqnarray
```

If not numbering equations, just create the contents for MATHJAX:

```
8904      \LWR@addmathjax{eqnarray*}{\BODY}
8905    }% LWR@numbreqnarray
8906 }% mathjax
8907 {% not mathjax
8908      \ifbool{LWR@numbreqnarray}
8909    {
```

For numbered svg equations, first create a lateximage with an alt attribute containing sanitized copy of the source code:

```
8910      \begin{BlockClass}{displaymathnumbered}%
8911      \LWR@newautoidanchor%
8912      \booltrue{LWR@indisplaymathimage}%
8913      \begin{lateximage}[{\LWR@startingequationtag--\LWR@equationtag}
8914        \LWR@addmathjax{eqnarray*}{\BODY}]
```

Then create the image contents using an actual eqnarray:

```
8915      \LWR@origeqnarray
8916      \BODY
8917      \LWR@origendeqnarray
8918      \end{lateximage}
8919      \end{BlockClass}
8920    }%
8921    {% not LWR@numbreqnarray
```

If not numbered, do the same, but an extra \nonumber seems to be required:

```
8922      \begin{BlockClass}{displaymath}
8923      \LWR@newautoidanchor%
8924      \booltrue{LWR@indisplaymathimage}%
8925      \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
8926      \LWR@origeqnarray
8927      \BODY
8928      \nonumber
8929      \LWR@origendeqnarray
8930      \end{lateximage}
8931      \end{BlockClass}
8932    }% LWR@numbreqnarray
8933 }% not mathjax
```

Default to number equations in the future:

```
8934 \booltrue{LWR@numbreqnarray}
8935 }
```

eqnarray itself is made with a blank line before and after to force it to be on its own line:

```
8936 \RenewEnviron{eqnarray}
8937 {%
8938
8939 \LWR@eqnarrayfactor
8940
8941 }
```

The starred version is patched to turn off the numbering:

```
8942 \csgpreto{eqnarray*}{\boolfalse{LWR@numbreqnarray}}
```

The following \mathcal{AMS} environments are more easily patched in-place:

Env `multline`

```
8943 \BeforeBeginEnvironment{multline}{
8944
8945 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8946 {
8947   \LWR@syncmathjax
8948   \booltrue{LWR@amsmultline}
8949   \LWR@beginhideamsmath
8950 }
8951 {
8952   \begin{BlockClass}{displaymathnumbered}
8953   \LWR@newautoidanchor%
8954   \booltrue{LWR@indisplaymathimage}%
8955   \begin{lateximage}[\LWR@amsmathbodynumbered{multline}]
8956 }
8957 }
8958
8959 \AfterEndEnvironment{multline}{
8960
8961 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8962 {
8963   \LWR@endhideamsmath
8964   \boolfalse{LWR@amsmultline}
8965   \LWR@addmathjax{multline}{\the\@envbody}
8966 }
8967 {\end{lateximage}\end{BlockClass}}
```

```

8968
8969 }

```

Env **multline***

```

8970 \BeforeBeginEnvironment{multline*}{
8971
8972 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8973 {
8974     \LWR@syncmathjax
8975     \booltrue{LWR@amsmultline}
8976     \LWR@beginhideamsmath
8977 }
8978 {
8979     \begin{BlockClass}{displaymath}
8980     \LWR@newautoidanchor
8981     \booltrue{LWR@indisplaymathimage}%
8982     \begin{lateximage}[\LWR@amsmathbody{multline*}]
8983 }
8984 }
8985
8986 \AfterEndEnvironment{multline*}{
8987
8988 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
8989 {
8990     \LWR@endhideamsmath
8991     \boolfalse{LWR@amsmultline}
8992     \LWR@addmathjax{multline*}{\the\@envbody}
8993 }
8994 {\end{lateximage}\end{BlockClass}}
8995
8996 }
8997

```

Env **gather**

```

8998 \BeforeBeginEnvironment{gather}{
8999
9000 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9001 {
9002     \LWR@syncmathjax
9003     \boolfalse{LWR@amsmultline}
9004     \LWR@beginhideamsmath
9005 }
9006 {
9007     \begin{BlockClass}{displaymathnumbered}
9008     \LWR@newautoidanchor%
9009     \booltrue{LWR@indisplaymathimage}%

```

```

9010 \begin{lateximage}[\LWR@amsmathbodynumbered{gather}]
9011 }
9012 }
9013
9014 \AfterEndEnvironment{gather}{
9015
9016 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9017 {
9018 \LWR@endhideamsmath
9019 \LWR@addmathjax{gather}{\the\@envbody}
9020 }
9021 {\end{lateximage}\end{BlockClass}}
9022
9023 }

```

Env **gather***

```

9024 \BeforeBeginEnvironment{gather*}{
9025
9026 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9027 {
9028 \LWR@syncmathjax
9029 \boolfalse{LWR@amsmultline}
9030 \LWR@beginhideamsmath
9031 }
9032 {
9033 \begin{BlockClass}{displaymath}
9034 \LWR@newautoidanchor%
9035 \booltrue{LWR@indisplaymathimage}%
9036 \begin{lateximage}[\LWR@amsmathbody{gather*}]
9037 }
9038 }
9039
9040 \AfterEndEnvironment{gather*}{
9041
9042 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9043 {
9044 \LWR@endhideamsmath
9045 \LWR@addmathjax{gather*}{\the\@envbody}
9046 }
9047 {\end{lateximage}\end{BlockClass}}
9048
9049 }

```

Env **align**

```

9050 \BeforeBeginEnvironment{align}{
9051

```

```

9052 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9053 {
9054     \LWR@syncmathjax
9055     \boolfalse{LWR@amsmultline}
9056     \LWR@beginhideamsmath
9057 }
9058 {
9059     \begin{BlockClass}{displaymathnumbered}
9060     \LWR@newautoidanchor%
9061     \booltrue{LWR@indisplaymathimage}%
9062     \begin{lateximage}[\LWR@amsmathbodynumbered{align}]
9063 }
9064 }
9065
9066 \AfterEndEnvironment{align}{
9067
9068 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9069 {
9070     \LWR@endhideamsmath
9071     \LWR@addmathjax{align}{\the\@envbody}
9072 }
9073 {\end{lateximage}\end{BlockClass}}
9074
9075 }

```

Env align*

```

9076 \BeforeBeginEnvironment{align*}{
9077
9078 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9079 {
9080     \LWR@syncmathjax
9081     \boolfalse{LWR@amsmultline}
9082     \LWR@beginhideamsmath
9083 }
9084 {
9085     \begin{BlockClass}{displaymath}
9086     \LWR@newautoidanchor%
9087     \booltrue{LWR@indisplaymathimage}%
9088     \begin{lateximage}[\LWR@amsmathbody{align*}]
9089 }
9090 }
9091
9092 \AfterEndEnvironment{align*}{
9093
9094 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9095 {
9096     \LWR@endhideamsmath
9097     \LWR@addmathjax{align*}{\the\@envbody}

```

```

9098 }
9099 {\end{lateximage}\end{BlockClass}}
9100
9101 }

```

Env **flalign**

```

9102 \BeforeBeginEnvironment{flalign}{
9103
9104 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9105 {
9106     \LWR@syncmathjax
9107     \boolfalse{LWR@amsmultline}
9108     \LWR@beginhideamsmath
9109 }
9110 {
9111     \begin{BlockClass}{displaymathnumbered}
9112     \LWR@newautoidanchor%
9113     \booltrue{LWR@indisplaymathimage}%
9114     \begin{lateximage}[\LWR@amsmathbodynumbered{flalign}]
9115 }
9116 }
9117
9118 \AfterEndEnvironment{flalign}{
9119
9120 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9121 {
9122     \LWR@endhideamsmath
9123     \LWR@addmathjax{flalign}{\the\@envbody}
9124 }
9125 {\end{lateximage}\end{BlockClass}}
9126
9127 }

```

Env **flalign***

```

9128 \BeforeBeginEnvironment{flalign*}{
9129
9130 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9131 {
9132     \LWR@syncmathjax
9133     \boolfalse{LWR@amsmultline}
9134     \LWR@beginhideamsmath
9135 }
9136 {
9137     \begin{BlockClass}{displaymath}
9138     \LWR@newautoidanchor%
9139     \booltrue{LWR@indisplaymathimage}%

```

```

9140 \begin{lateximage}[\LWR@amsmathbody{flalign*}]
9141 }
9142 }
9143
9144 \AfterEndEnvironment{flalign*}{
9145
9146 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9147 {
9148 \LWR@endhideamsmath
9149 \LWR@addmathjax{flalign*}{\the\@envbody}
9150 }
9151 {\end{lateximage}\end{BlockClass}}
9152
9153 }

```

Env alignat

```

9154 \BeforeBeginEnvironment{alignat}{
9155
9156 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9157 {
9158 \LWR@syncmathjax
9159 \boolfalse{LWR@amsmultline}
9160 \LWR@beginhideamsmath
9161 }
9162 {
9163 \begin{BlockClass}{displaymathnumbered}
9164 \LWR@newautoidanchor%
9165 \booltrue{LWR@indisplaymathimage}%
9166 \begin{lateximage}[\LWR@amsmathbodynumbered{alignat}]
9167 }
9168 }
9169
9170 \AfterEndEnvironment{alignat}{
9171
9172 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9173 {
9174 \LWR@endhideamsmath
9175 \LWR@addmathjax{alignat}{\the\@envbody}
9176 }
9177 {\end{lateximage}\end{BlockClass}}
9178
9179 }

```

Env alignat*

```

9180 \BeforeBeginEnvironment{alignat*}{
9181

```

```

9182 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9183 {
9184   \LWR@syncmathjax
9185   \boolfalse{LWR@amsmultline}
9186   \LWR@beginhideamsmath
9187 }
9188 {
9189   \begin{BlockClass}{displaymath}
9190   \LWR@newautoidanchor%
9191   \booltrue{LWR@indisplaymathimage}%
9192   \begin{lateximage}[\LWR@amsmathbody{alignat*}]
9193 }
9194 }
9195
9196 \AfterEndEnvironment{alignat*}{
9197
9198 \ifboolexpr{bool{mathjax} or ( bool{FormatWP} and bool{WPMarkMath} ) }%
9199 {
9200   \LWR@endhideamsmath
9201   \LWR@addmathjax{alignat*}{\the\@envbody}
9202 }
9203 {\end{lateximage}\end{BlockClass}}
9204
9205 }

9206 \end{warpHTML}

```

76 Lateximages

76.1 Description

Env `lateximage` A `lateximage` is a piece of the document which is typeset in \TeX then included in the HTML output as an image. This is used for math if `svg math` is chosen, and also for the `picture`, `tikzpicture`, and other environments.

Before typesetting the `lateximage` a large number of formatting, graphics, and symbols-related macros are temporarily restored to their print-mode meaning by `\LWR@restoreorigformatting`. (See section 74.)

A `lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceeding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML `` is created to hold both the HTML comment, which will have the **pdftotext** conversion, and also the link to the final `.svg` image.

A \LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

HTML alt tag The HTML alt tag is set to the \LaTeX source for SVG math, some chemistry expressions, and perhaps some other expressions which make sense for text copy/paste. In some other cases, the alt tag is set according to the package name.

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

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

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

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

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

SVG image font size For the `lateximage` environment, the size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, which defaults to:

```
\renewcommand{\LateximageFontSizeName}{normalsize}
```

For inline SVG math, font size is instead controlled by `\LateximageFontScale`, which defaults to:

```
\newcommand*{\LateximageFontScale}{.75}
```

76.2 Support counters and macros

for HTML output: 9207 `\begin{warpHTML}`

Ctr LWR@lateximagenumber Sequence the images.

```
9208 \newcounter{LWR@lateximagenumber}
9209 \setcounter{LWR@lateximagenumber}{0}
```

Ctr LWR@lateximagedepth Do not create \lateximage inside of \lateximage.

```
9210 \newcounter{LWR@lateximagedepth}
9211 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
9212 \edef\LWR@hashmark{\string#} % for use in \write
9213 \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.


```
9214 \newcounter{LWR@LIpage}

9215 \end{warpHTML}
```

76.3 Font size

for HTML & PRINT: 9216 \begin{warpall}

\LateximageFontSizeName Declares how large to write text in \lateximages. The .svg file text size should blend well with the surrounding HTML text size.

 **no backslash** *Do not include the leading backslash in the name.*

```
9217 \newcommand*{\LateximageFontSizeName}{normalsize}
```

\LateximageFontScale Declares how large to scale inline svg math images. The .svg file text size should blend well with the surrounding HTML text size. The default is .75, but it may be redefined as needed depending on the HTML font.

```
9218 \newcommand*{\LateximageFontScale}{.75}
```

```
9219 \end{warpall}
```

76.4 Sanitizing math expressions for HTML

for HTML output: 9220 `\begin{warpHTML}`

`\LWR@HTMLsanitize` $\{\langle text \rangle\}$

Math expressions are converted to `lateximages`, and some math environments may contain `&`, `<`, or `>`, which should not be allowed inside an HTML `<alt>` tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```
9221 \newrobustcmd{\LWR@HTMLsanitize}[1]{%
```

Cancel French **babel** character handling, and fully expand the strings:

```
9222 \begingroup%
9223 \LWR@FBcancel%
9224 \fullexpandarg%
```

The `&`, `<`, and `>` may be interpreted by the browser:

```
9225 \protect\StrSubstitute{\detokenize{#1}}%
9226 {\detokenize{&}}{\detokenize{&}}[\LWR@strresult]%

9227 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9228 {\detokenize{<}}{\detokenize{&lt;}}[\LWR@strresult]%

9229 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9230 {\detokenize{>}}{\detokenize{&gt;}}[\LWR@strresult]%
```

The double quote occasionally causes problems.

```
9231 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9232 {\detokenize{"}}{\detokenize{&quot;}}[\LWR@strresult]%
```

MathJax allows expressions to be defined with `\newcommand`. These expressions would appear with `##` for each argument, and each must be changed to a single `#`. This must be done after all the above changes. Attempting another conversion after this causes an error upon further expansion.

```
9233 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9234 {\detokenize{##}}{\LWR@origpound}[\LWR@strresult]%

9235 \LWR@strresult%
9236 \endgroup%
9237 }
```

`\LWR@HTMLsanitizeexpand` $\{\langle text \rangle\}$

This version expands the argument before sanitizing it.

```
9238 \newrobustcmd{\LWR@HTMLsanitizeexpand}[1]{%
```

Cancel French **babel** character handling, and fully expand the strings:

```
9239 \begingroup%
9240 \LWR@FBcancel%
9241 \fullexpandarg%
```

The difference between this and `\LWR@HTMLsanitize` (without “expand”) is the following `\expandafter`:

```
9242 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
9243 {\detokenize{&}}{\detokenize{&}}[\LWR@strresult]%

9244 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9245 {\detokenize{<}}{\detokenize{&lt;}}[\LWR@strresult]%

9246 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9247 {\detokenize{>}}{\detokenize{&gt;}}[\LWR@strresult]%

9248 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
9249 {\detokenize{"}}{\detokenize{&quot;}}[\LWR@strresult]%
```

`\LWR@HTMLsanitizeexpand` is not used for defining new MathJax macros, so the `##` conversion is not needed here.

```
9250 \LWR@strresult%
9251 \endgroup%
9252 }
```

76.5 Equation numbers

`\LWR@startingequation` For use with `lateximage` and multi-line numbered equations. Remembers the next equation number so that it may be printed in the alt tag.

```
9253 \newcounter{\LWR@startingequation}
9254
9255 \@ifundefined{chapter}
9256 {
9257 \renewcommand{\the\LWR@startingequation}{%
```

```

9258 \arabic{LWR@startingequation}%
9259 }
9260 }
9261 {% chapter defined
9262 \renewcommand{\theLWR@startingequation}{%
9263 \ifnumcomp{\value{chapter}}{>}{0}{\arabic{chapter}.}{}}%
9264 \arabic{LWR@startingequation}%
9265 }
9266 }

```

Bool True for the first equation tag, false for later tags in the same environment.

LWR@isstartingequation

```
9267 \newbool{LWR@isstartingequation}
```

\LWR@startingequationtag Prints the starting equation number or tag.

```
9268 \let\LWR@startingequationtag\theLWR@startingequation
```

\LWR@equationtag Prints the ending equation number or tag.

This is reset by lateximage, may be temporarily overwritten by \tag calling \LWR@remembertag.

```
9269 \newcommand*{\LWR@equationtag}{}
```

Only if svg math, patch \tag after packages have loaded, in case someone else modified \tag.

```

9270 \AtBeginDocument{
9271
9272 \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.

```

9273 \NewDocumentCommand{\LWR@remembertag}{m}{%
9274 \ifbool{LWR@isstartingequation}%
9275 {%
9276   \global\boolfalse{LWR@isstartingequation}%
9277   \xdef\LWR@startingequationtag{#1}%
9278 }{}}%
9279 \xdef\LWR@equationtag{#1}%
9280 }%

```

Patches for \mathcal{AMS} math `\tag` macro to remember the first tag:

```

9281 \LetLtxMacro\LWR@origmake@df@tag@@\make@df@tag@@
9282 \LetLtxMacro\LWR@origmake@df@tag@@@\make@df@tag@@@
9283
9284 \renewcommand*\make@df@tag@@[1]{%
9285 \LWR@remembertag{#1}%
9286 \LWR@origmake@df@tag@@{#1}%
9287 }
9288
9289 \renewcommand*\make@df@tag@@@[1]{%
9290 \LWR@remembertag{#1}%
9291 \LWR@origmake@df@tag@@@{#1}%
9292 }
9293
9294 }% not mathjax
9295 }% AtBeginDocument

```

76.6 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.

```

9296 \newcommand*\LWR@amsmathbody[1]
9297 {%
9298 \textbackslash\begin\}\{#1\} % extra space
9299 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}%
9300 \textbackslash\end\}\{#1\}%
9301 }

```

`\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.

```

9302 \newcommand*\LWR@amsmathbodynumbered[1]
9303 {%
9304 \ifnumcomp{\value{LWR@startingequation}}{=}{\value{equation}}%
9305 {(\LWR@equationtag)}%
9306 {(\LWR@startingequationtag--\LWR@equationtag)} % extra space
9307 \LWR@amsmathbody{#1} % extra space
9308 }

```

76.7 lateximage environment

Env `lateximage` * [`<2: <alt> tag`] [`<3: add'l hashing`] [`<4: CSS style`]

Typesets the contents and then renders the result as an svg file. Star causes the image to be hashed for reuse.

The optional `<alt>` tag is included in the HTML code for use with copy/paste.

image filename hashing If starred, a hashed filename is used. If so, the hash is based on the `alt` tag and also the additional hashing argument.

This may be used to provide an expression with a simple `alt` tag but also enough additional information to provide a unique hash.

An example is when the expression is a complicated \TeX expression, which would not copy/paste well. A simplified tag may be used, while the complicated expression is duplicated in the additional hashing argument.

Another example is when the expression is simple, but the image depends on options. These options may be decoded into text form and included in the additional hashing argument in order to make the hash unique according to the set of options, even if the simple `alt` tag is still the same.

```
9309 \catcode'\$=\active%
9310
9311 \NewDocumentEnvironment{lateximage}{s O{(image)} O{} O{}}
9312 {%
9313 \LWR@traceinfo{lateximage: starting on \jobname.pdf page \arabic{page}}%
9314 \LWR@traceinfo{lateximage: entering depth is \arabic{LWR@lateximagedepth}}%
```

Nested lateximages remain one large lateximage:

```
9315 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%
```

If nesting inside an already-existing lateximage, simply record one more level. \mathcal{AMS} packages redefine `\addtocounter` to do nothing if inside a `\text`, so lower-level \TeX macros are used for tracking nested lateximages.

```
9316 {%
9317 % \addtocounter{LWR@lateximagedepth}{1}%
9318 \global\advance\c@LWR@lateximagedepth 1\relax% Due to AmS \text macro.
9319 }%
```

Otherwise, this is the outer-most lateximage:

```
9320 {% start of outer-most lateximage
```

Remember the next equation number to be allocated, in case it must be printed in a multi-equation environment:

```
9321 \LWR@traceinfo{lateximage: starting outer-most lateximage}%
9322   \setcounter{LWR@startingequation}{\value{equation}}%
9323   \addtocounter{LWR@startingequation}{1}%
9324   \booltrue{LWR@isstartingequation}%
9325   \let\LWR@startingequationtag\theLWR@startingequation%
```

The default equation tag, unless overwritten by `\tag`:

```
9326   \let\LWR@equationtag\theequation%
```

Starting a new lateximage:

```
9327   \addtocounter{LWR@lateximagenumber}{1}%
9328   \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
9329   \boolfalse{mathjax}%
```

Be sure that are doing a paragraph:

```
9330   \LWR@ensuredoingapar%
```

Next file:

```
9331   \addtocounter{LWR@externalfilecnt}{1}%
9332   \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`:

```
9333   \setcounterpageref{LWR@LIpage}{LWRlateximage\arabic{LWR@lateximagenumber}}%
9334   \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:

```
9335   \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" % extra space
9336   class="lateximagesource"}%
```

Write instructions to the `lateximages.txt` file:

```
9337   \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
9338   \IfBooleanTF{#1}% starred
```

```

9339    {% hash
9340        \LWR@traceinfo{lateximage: hash true, adding %
9341            !\detokenize\expandafter{#2}!\detokenize\expandafter{#3}!}%

```

Compute and save the hashed file name for later use:

```

9342        \edef\LWR@hashedname{%
9343            \LWR@mdfive{\detokenize\expandafter{#2}-!-#3}%
9344    %        \LWR@mdfive{\detokenize\expandafter{#2}-!-\detokenize\expandafter{#3}}}%
9345        }%
9346        \LWR@traceinfo{lateximage: hash is \LWR@hashedname}%

```

Write the page, hashing, and hashed name:

```

9347        \immediate\write\LWR@lateximagesfile{%
9348            |\arabic{LWR@LPage}|true|\LWR@hashedname|
9349        }%
9350    }% hash
9351    {% no hash

```

No hash, so write the page, no hashing, and the image number:

```

9352        \LWR@traceinfo{lateximage: hash false}%
9353        \immediate\write\LWR@lateximagesfile{%
9354            |\arabic{LWR@LPage}|false|lateximage-\arabic{LWR@externalfilecnt}|
9355        }%
9356    }% no hash

```

Place an open comment tag. This will hide any traces of the lateximage PDF page which were picked up by **pdftotext**.

```

9357    \LWR@traceinfo{lateximage: about to create open comment}%
9358    \LWR@htmlopencomment%

```

One level deeper. At this outer-most lateximage, it is known that this is not being used inside an \mathcal{MS} \text, since the outer-most level will never be in math mode.

```

9359    \addtocounter{LWR@lateximagedepth}{1}%

```

Start the new PDF page:

```

9360    \LWR@traceinfo{lateximage: about to create a new page}%
9361    \LWR@originewpage%

```

Typeset the image in a “standard” width page and font size:

```

9362    \LWR@traceinfo{lateximage: about to create minipage}%
9363    \LWR@print@minipage{6in}%

```

```
9364 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a `lateximage`.

```
9365 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
9366 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
9367 \def\@mpfn{footnote}%
9368 \def\thempfn{\thefootnote}%
9369 \LetLtxMacro\@footnotetext\LWR@footnotetext%
```

Create the `LWRlateximage<number>` label:

```
9370 \LWR@traceinfo{lateximage: about to create label}%
9371 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
9372 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
9373 \LetLtxMacro$\LWR@origdollar%
9374 \catcode'\$=3% math shift
9375 \LetLtxMacro\(\LWR@origopenparen%
9376 \LetLtxMacro\)\LWR@origcloseparen%
```

Only enable print-mode display math if are not already inside display math:

```
9377 \ifbool{LWR@indisplaymathimage}{}{% not in display math
9378     \LetLtxMacro\[\LWR@origopenbracket%
9379     \LetLtxMacro\]\LWR@origclosebracket%
9380     \let\equation\LWR@origequation%
9381     \let\endequation\LWR@origendequation%
9382     \csletcs{equation*}{LWR@origequation*}%
9383     \csletcs{endequation*}{LWR@origendequation*}%
9384 }% not in display math
```

For **chemformula**:

```
9385 \LetLtxMacro\LWR@newsingledollar$%
9386 \LetLtxMacro\LWR@newsingledollar$% syntax highlighting

9387}% end of outer-most lateximage
9388 \LWR@traceinfo{lateximage: finished start of environment}%
9389}% end of \begin{lateximage}
```

`\endlateximage` When the environment closes:

```
9390 {% start of \end{lateximage}
9391 \LWR@traceinfo{lateximage: starting end of lateximage}%
```

Nested more than one deep?

```
9392 \LWR@traceinfo{lateximage: internal depth was \arabic{LWR@lateximagedepth}}%
9393 \ifnumcomp{\value{LWR@lateximagedepth}}{>}{1}%
```

If nesting inside an already existing lateximage, simply record one less level. Uses a lower-level \TeX macro due to $\mathcal{A}\mathcal{M}\mathcal{S}$ `\text` change of `\addtocounter`.

```
9394 {%
9395   \LWR@traceinfo{lateximage: unnesting}%
9396   \global\advance\c@LWR@lateximagedepth -1\relax%
9397 }%
```

If this is the outer-most lateximage:

```
9398 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
9399 \LWR@traceinfo{lateximage: ending outer-most lateximage}%
9400   \endLWR@print@minipage%
9401   \LWR@orignewpage%
9402   \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by **pdftotext**:

```
9403   \LWR@print@vspace*{.5\baselineskip}%
9404   \LWR@htmlclosecomment%
9405   \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

```
9406   \IfBooleanTF{#1}% starred
9407   {% hash
9408     \LWR@subinlineimage[#2]{lateximage}%
9409     {%
9410       lateximages\OSPathSymbol%
9411       \LWR@print@mbbox{\LWR@hashedname}%
9412     }{svg}{#4}%
9413   }% hash
9414   {% no hash
9415     \LWR@subinlineimage[#2]{lateximage}%
```

```

9416      {%
9417          lateximages\OSPathSymbol%
9418          \LWR@print@mbbox{lateximage-\theLWR@externalfilecnt}%
9419      }{svg}{#4}%
9420  }% no hash

```

Be sure that are doing a paragraph:

```

9421  \LWR@ensuredoingapar%

```

Close the HTML span which has the **pdfotext** comment and also the link to the .svg image:

```

9422  \LWR@htmltag{/span}%
9423  \ifbool{HTMLDebugComments}{%
9424      \LWR@htmlcomment{End of lateximage}%
9425  }{}%

```

Undo one lateximage level. This is not inside an \AA MS \text, so regular \addtocounter may be used here.

```

9426  \addtocounter{LWR@lateximagedepth}{-1}%
9427 }% end of outer-most lateximage
9428 \LWR@traceinfo{lateximage: exiting depth is \arabic{LWR@lateximagedepth}}%
9429 \LWR@traceinfo{lateximage: done}%
9430 }%
9431 \catcode'\$=3% math shift
9432 \end{warpHTML}

```

for PRINT output: 9433 \begin{warpprint}

Env lateximage [$\langle alt \rangle$ tag] [$\langle CSS style \rangle$]

varwidth is used to create a box of the natural width of its contents.

```

9434 \NewDocumentEnvironment{lateximage}{s o o o}
9435     {\begin{varwidth}[b]{\linewidth}}
9436     {\end{varwidth}}

9437 \end{warpprint}

```

77 center, flushleft, flushright

for HTML output: 9438 \begin{warpHTML}

Env **center** Replace center functionality with CSS tags:

```

9439 \renewenvironment*{center}
9440 {
9441 \LWR@forcenewpage
9442 \ifbool{FormatWP}
9443 {\BlockClass[\LWR@print@mbox{text-align:center}]{center}}
9444 {\BlockClass{center}}
9445 }
9446 {\endBlockClass}

```

Env **flushright**

```

9447 \renewenvironment*{flushright}
9448 {
9449 \LWR@forcenewpage
9450 \ifbool{FormatWP}
9451 {\BlockClass[\LWR@print@mbox{text-align:right}]{flushright}}
9452 {\BlockClass{flushright}}
9453 }
9454 {\endBlockClass}

```

Env **flushleft**

```

9455 \renewenvironment*{flushleft}
9456 {
9457 \LWR@forcenewpage
9458 \ifbool{FormatWP}
9459 {\BlockClass[\LWR@print@mbox{text-align:left}]{flushleft}}
9460 {\BlockClass{flushleft}}
9461 }
9462 {\endBlockClass}

```

`\centering`, `\raggedleft`, and `\raggedright` usually have no effect on the HTML output, but they may be used to compare with the next token to identify their use at the start of a float. See `\LWR@floatalignment`.

\centering

```

9463 \renewcommand*{\centering}{%
9464 \ifbool{HTMLDebugComments}{%
9465   \LWR@htmlcomment{centering}%
9466 }{}%
9467 }

```

`\raggedleft`

```
9468 \renewcommand*{\raggedleft}{%
9469 \ifbool{HTMLDebugComments}{%
9470     \LWR@htmlcomment{raggedleft}%
9471 }{}%
9472 }
```

`\raggedright`

```
9473 \renewcommand*{\raggedright}{%
9474 \ifbool{HTMLDebugComments}{%
9475     \LWR@htmlcomment{raggedright}%
9476 }{}%
9477 }
```

`\leftline` $\{\langle text \rangle\}$

```
9478 \renewcommand{\leftline}[1]{\begin{flushleft}#1\end{flushleft}}
```

`\centerline` $\{\langle text \rangle\}$

```
9479 \renewcommand{\centerline}[1]{\begin{center}#1\end{center}}
```

`\rightline` $\{\langle text \rangle\}$

```
9480 \renewcommand{\rightline}[1]{\begin{flushright}#1\end{flushright}}
```

```
9481 \end{warpHTML}
```

78 Pre-loaded packages

for HTML output: 9482 \begin{warpHTML}

If **textcomp** was loaded before **lwarp**, perhaps as part of the font-related packages, explicitly load the lwarp patches now:

```
9483 \@ifpackageloaded{textcomp}
9484 {
9485 \LWR@origRequirePackage{lwarp-textcomp}
9486 }
9487 {}
```

If **graphics** or **graphicx** were loaded before **lwarp**, perhaps by **xunicode**, explicitly load the lwarp patches now:

```

9488 \@ifpackageloaded{graphics}
9489 {
9490 \LWR@origRequirePackage{lwarp-graphics}
9491 }
9492 {}

9493 \end{warpHTML}

```


79 Siunitx

Pkg **siunitx** The **lwarp** core passes a few options to **siunitx**.

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

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

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

 **tabular** Tabular S columns are rendered as simple c columns, and tabular s columns are not supported. These may be replaced by c columns with each cell contained in `\num` or `\si`.

for HTML output: 9494 \begin{warpHTML}

Options for siunitx:

```

9495 \newrobustcmd{\LWR@siunitx@textcelsius}{\HTMLentity{deg}C}
9496 \newrobustcmd{\LWR@siunitx@textdegree}{\HTMLentity{deg}}
9497 \newrobustcmd{\LWR@siunitx@textprime}{\HTMLUnicode{2032}}
9498 \newrobustcmd{\LWR@siunitx@textdblprime}{\HTMLUnicode{2033}}
9499 \newrobustcmd{\LWR@siunitx@textplanckbar}{\text{\textit{\HTMLUnicode{0127}}}}
9500
9501 \appto\LWR@restoreorigformatting{%
9502 \renewrobustcmd{\LWR@siunitx@textcelsius}{\text{\ensuremath{^\circ}C}}%
9503 \renewrobustcmd{\LWR@siunitx@textdegree}{\text{\ensuremath{^\circ}}}%
9504 \renewrobustcmd{\LWR@siunitx@textprime}{\text{\ensuremath{^\prime}}}%
9505 \renewrobustcmd{\LWR@siunitx@textdblprime}{\text{\ensuremath{^\prime\prime}}}%
9506 \renewrobustcmd{\LWR@siunitx@textplanckbar}{\text{\ensuremath{\hbar}}}%
9507 }
9508

```

```

9509 \PassOptionsToPackage{
9510     detect-mode=true,
9511     per-mode=symbol,% fraction is not seen by pdftotext
9512     text-celsius = {\LWR@siunitx@textcelsius},
9513     text-degree = {\LWR@siunitx@textdegree},
9514     text-arcminute = {\LWR@siunitx@textprime} ,
9515     text-arcsecond = {\LWR@siunitx@textdblprime} ,
9516 }{siunitx}

9517 \end{warpHTML}

```

80 Graphics print-mode modifications

80.1 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, or 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
```

For a large number of images, use **lwarpmk**:

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

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. Do not use the `scale` option, since it is not well supported by HTML browsers.

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

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

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

⚠ **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The css3 standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \TeX , so expect some ugly results for scaling and rotating.

80.2 Print-mode modifications

for PRINT output: For print output, accept and then discard the new class key:

```
9518 \begin{warpprint}
9519 \define@key{Gin}{class}{}

```

Print-mode additions for the **overpic** package. See section 274 for the HTML version.

```
9520 \AtBeginDocument{
9521 \@ifpackageloaded{overpic}{
9522 \newcommand*\overpicfontsize{12}
9523 \newcommand*\overpicfontskip{14}
9524 }{}
9525 }
9526 \end{warpprint}

```

81 Xcolor boxes

Pkg **xcolor** A few new definitions are provided for enhanced HTML colored boxes, and `\fcolorbox`

is slightly modified. Print-mode version are also provided.

Print-mode versions of new **xcolor** 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: 9527 `\begin{warpall}`

After **xparse** may have been loaded ...

9528 `\AtBeginDocument{`

... and *only* if **xcolor** was loaded:

9529 `\@ifpackageloaded{xcolor}{`
9530 `\LWR@traceinfo{patching xcolor}`

The print version:

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

9531 `\LetLtxMacro\colorboxBlock\colorbox`

The original definition is reused by the new versions:

9532 `\LetLtxMacro\LWR@orig@print@fcolorbox\fcolorbox`

`\fcolorbox` [`\framemodel`] [`\framecolor`] [`\boxmodel`] [`\boxcolor`] [`\text`]

In print mode, `\fcolorbox` is modified to accept a background color of `none`.

(`\fcolorbox` is particular about its optional arguments, thus the elaborate combinations of `\ifthenelse`.)

9533 `\newsavebox{\LWR@colorminipagebox}`
9534
9535 `\NewDocumentCommand{\LWR@print@fcolorbox}{o m o m +m}{%`
9536 `\LWR@traceinfo{\LWR@print@fcolorbox #2 #4}%`

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

9537 `\begin{lrbox}{\LWR@colorminipagebox}%`
9538 `#5%`
9539 `\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.

```

9540 \ifthenelse{\equal{#4}{none}}{%
9541 {% #4 none
9542   \LWR@traceinfo{background is none}%
9543   {% scope the \colorlet
9544     \colorlet{LWR@currentcolor}{.}%
9545     \color{#2}%
9546     \fbox{%
9547       \color{LWR@currentcolor}%
9548       \usebox{\LWR@colorminipagebox}%
9549     }% fbox
9550   }% colorlet
9551 }% #4 none
9552 {% #4 not none
9553   \LWR@traceinfo{background not none}%
9554   \IfValueTF{#1}%
9555   {%
9556     \IfValueTF{#3}%
9557     {\LWR@orig@print@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
9558     {\LWR@orig@print@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
9559   }%
9560   {% no value #1
9561     \IfValueTF{#3}%
9562     {\LWR@orig@print@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
9563     {\LWR@orig@print@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
9564   }% no value #1
9565   }% #4 not none
9566   \LWR@traceinfo{LWR@print@fcolorbox done}%
9567 }
9568
9569 \renewcommand*\fcolorbox{\LWR@print@fcolorbox}

```

`\fcolorboxBlock` [*<framemodel>*] [*<framecolor>*] [*<boxmodel>*] [*<boxcolor>*] [*<text>*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```

9570 \newcommand*\LWR@print@fcolorboxBlock{\LWR@print@fcolorbox}
9571 \newcommand*\fcolorboxBlock{\LWR@print@fcolorboxBlock}

```

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:

```
9572 \NewDocumentEnvironment{LWR@print@fcolorminipage}{o m o m O{c} O{ } o m}
9573 {%
9574 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%
```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
9575 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
9576 \IfValueTF{#7}%
9577 {\begin{minipage}[#5][#6][#7]{#8}}%
9578 {\begin{minipage}[#5][#6][#5]{#8}}%
9579 }%
9580 {%
9581 \end{minipage}%
9582 \end{lrbox}%
9583 \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.

```
9584 \ifthenelse{\equal{#4}{none}}%
9585 {% #4 none
9586   {% scope the \colorlet
9587     \colorlet{LWR@currentcolor}{.}%
9588     \color{#2}%
9589     \fbox{%
9590       \color{LWR@currentcolor}%
9591       \usebox{\LWR@colorminipagebox}%
9592     }% fbox
9593   }% colorlet
9594 }% #4 none
9595 {% #4 not none
9596   \IfValueTF{#1}%
9597   {%
9598     \IfValueTF{#3}%
9599     {\LWR@orig@print@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
9600     {\LWR@orig@print@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
9601   }%
9602   {% no value #1
9603     \IfValueTF{#3}%
9604     {\LWR@orig@print@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
9605     {\LWR@orig@print@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%

```

```

9606     }% no value #1
9607 }% #4 not none
9608 \LWR@traceinfo{*** finished end fcolorminipage}%
9609 }
9610
9611 \newenvironment*{fcolorminipage}
9612     {\LWR@print@fcolorminipage}
9613     {\endLWR@print@fcolorminipage}

9614 \LWR@traceinfo{xcolor patches done}
9615 }{}% xcolor loaded
9616 }% AtBeginDocument

9617 \end{warpall}

```

82 Chemmacros environments

`\makepolymerdelims` and redox reactions must be enclosed in a `lateximage` during HTML output. These environments are provided here in print mode, and in the **chemmacros** code in HTML mode, as a high-level semantic syntax which automatically embeds the contents in a `lateximage` with an appropriate alt tag.

for PRINT output: 9618 `\begin{warpprint}`

```

9619 \AtBeginDocument{
9620 \ifpackageloaded{chemmacros}{

```

Env `polymerdelims`

```

9621 \DeclareDocumentEnvironment{polymerdelims}{}
9622     {}{}

```

Env `redoxreaction` `{\langle space above \rangle}` `{\langle space below \rangle}`

For print output, extra space is include above and below the image, and a `lateximage` is not necessary. This extra space must be enforced, even inside a float, so zero-width rules are used.

For the HTML version, see section [142.4](#).

```

9623 \DeclareDocumentEnvironment{redoxreaction}{m m}
9624     {\rule{0pt}{\#1}}{\rule[-\#2]{0pt}{\#2}}

9625 }{}% chemmacros
9626 }% AtBeginDocument

```

```
9627 \end{warpprint}
```

83 Cleveref

Pkg cleveref **cleveref** package is used as-is with minor patches.



cleveref page numbers

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

Ex:

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

See `\cpagerefFor` at page 513 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 425 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: 9628 `\begin{warppHTML}`

`\AtEndPreamble` forces **cleveref** to be loaded last:

```
9629 \AtEndPreamble{
9630 \RequirePackage{cleveref}
9631 }
```

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.

```

9632 \AfterEndPreamble{
9633 \LWR@traceinfo{Patching cleveref.}

```

`\@@@setcref` $\{\langle kindofref \rangle\} \{\langle label \rangle\}$

`\@templabel` becomes the section number.

```

9634 \def\LWR@orig@@@setcref#1#2{\cref@getlabel{#2}{\@templabel}#1{\@templabel}{}}%
9635
9636 \ifdefequal{\@@@setcref}{\LWR@orig@@@setcref}{% before v0.21
9637   \renewcommand*\@@@setcref}[2]{#1{\ref{#2}}{}}
9638 }{
9639   \ifdefequal{\@@@setcref}{\LWR@orig@@@setcref}{% as of v0.21
9640     \renewcommand*\@@@setcref}[2]{#1{\ref{#2}}{}}
9641   }{
9642     \PackageWarning{lwarp-cleveref}{
9643       Unknown version of cleveref.
9644       \protect\cref\space will fail.
9645     }%
9646   }
9647 }

```

`\@@@setcrefrange` $\{\langle text \rangle\} \{\langle label \rangle\} \{\langle label \rangle\}$

```

9648 \def\LWR@orig@@@setcrefrange#1#2#3{%
9649   \cref@getlabel{#2}{\@labela}%
9650   \cref@getlabel{#3}{\@labelb}%
9651   #1{\@labela}{\@labelb}{}}%
9652
9653 \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{
9654   \renewcommand{\@@@setcrefrange}[3]{%
9655     #1{\ref{#2}}{\ref{#3}}{}}%
9656 }
9657 }{
9658   \ifdefequal{\@@@setcrefrange}{\LWR@orig@@@setcrefrange}{
9659     \renewcommand{\@@@setcrefrange}[3]{%
9660       #1{\ref{#2}}{\ref{#3}}{}}%
9661     }
9662   }{
9663     \PackageWarning{lwarp-cleveref}{
9664       Unknown version of cleveref.
9665       \protect\crefrange\space will fail.
9666     }
9667   }
9668 }
9669

```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
9670 \newcommand*{\cpagerefFor}{for}
```

`\@@@setcpageref` $\{\langle typeofref \rangle\} \{\langle label \rangle\}$, where `typeofref` is “page” or “pages”

```
9671 \def\LWR@orig@@setcpageref#1#2{% before v0.21
9672   \cref@getpageref{#2}{\@temppage}#1{\@temppage}{-}{-}}%
9673
9674 \def\LWR@orig@@@setcpageref#1#2{% as of v0.21
9675   \cpageref@getlabel{#2}{\@temppage}#1{\@temppage}{-}{-}}%
9676
9677 \ifdefequal{\@@setcpageref}{\LWR@orig@@setcpageref}{
9678   \renewcommand*{\@@setcpageref}[2]{%
9679     #1{\cpagerefFor\ \cref{#2}}{-}{-}}%
9680   }
9681 }{
9682   \ifdefequal{\@@@setcpageref}{\LWR@orig@@@setcpageref}{
9683     \renewcommand*{\@@@setcpageref}[2]{%
9684       #1{\cpagerefFor\ \cref{#2}}{-}{-}}%
9685     }
9686   }
9687   {
9688     \PackageWarning{lwarp-cleveref}{
9689       Unknown version of cleveref.
9690       \protect\cpageref\space will fail.
9691     }
9692   }
9693 }

9694 \def\LWR@orig@@setcpagerefrange#1#2#3{% before v0.21
9695   \cref@getpageref{#2}{\@pagea}%
9696   \cref@getpageref{#3}{\@pageb}%
9697   #1{\@pagea}{\@pageb}{-}{-}{-}}%
9698
9699 \def\LWR@orig@@@setcpagerefrange#1#2#3{% as of v0.21
9700   \cpageref@getlabel{#2}{\@pagea}%
9701   \cpageref@getlabel{#3}{\@pageb}%
9702   #1{\@pagea}{\@pageb}{-}{-}{-}}%
9703
9704 \ifdefequal{\@@setcpagerefrange}{\LWR@orig@@setcpagerefrange}{
9705   \renewcommand*{\@@setcpagerefrange}[3]{%
9706     #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{-}}%
9707   }
9708 }{
9709   \ifdefequal{\@@@setcpagerefrange}{\LWR@orig@@@setcpagerefrange}{
9710     \renewcommand*{\@@@setcpagerefrange}[3]{%
9711       #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{-}}%
9712   }
9713 }
```

```

9712     }
9713   }
9714   {
9715     \PackageWarning{lwarp-cleveref}{
9716       Unknown version of cleveref.
9717       \protect\cpagerefrange\space will fail.
9718     }
9719   }
9720 }
9721
9722 }% AfterEndPreamble

```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```

9723 \LetLtxMacro\LWR@origlabel\label
9724 \RenewDocumentCommand{\label}{}{\LWR@newlabel}
9725
9726 \LetLtxMacro\LWR@origref\ref
9727 \RenewDocumentCommand{\ref}{}{\LWR@newref}%
9728
9729 \LetLtxMacro\LWR@origpageref\pageref
9730 \RenewDocumentCommand{\pageref}{}{\LWR@newpageref}
9731
9732 \end{warpHTML}

```

84 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 9733 `\begin{warpHTML}`

Env `picture`


```

9734 \BeforeBeginEnvironment{picture}{\begin{lateximage}}
9735
9736 \AfterEndEnvironment{picture}{\end{lateximage}}
9737
9737 \end{warpHTML}

```

85 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

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

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


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

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

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

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

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

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

for HTML output: 9738 `\begin{warphTML}`

85.1 Counters and lengths

Ctrl `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
9739 \newcounter{LWR@minipagedepth}
9740 \setcounter{LWR@minipagedepth}{0}
```

Len `\LWR@minipagewidth` Used to convert the width into printable units.

```
9741 \newlength{\LWR@minipagewidth}
```

Len \LWR@minipageheight Used to convert the height into printable units.

```
9742 \newlength{\LWR@minipageheight}
```

85.2 Footnote handling

Also see section 55 for other forms of footnotes. Minipage footnotes are gathered in section 55.5, and then placed into the document in section 85.3.

85.3 Minipage handling

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```
9743 \newbool{LWR@minipagefullwidth}
9744 \boolfalse{LWR@minipagefullwidth}
```

\minipagefullwidth Requests that the next minipage have no width tag in HTML:

for HTML output: 9745 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
9746 \end{warpHTML}

for PRINT output: 9747 \begin{warpprint}
9748 \newcommand*{\minipagefullwidth}{}
9749 \end{warpprint}

for HTML output: 9750 \begin{warpHTML}

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```
9751 \newbool{LWR@minipagethispar}
9752 \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.

```

9753 \NewDocumentEnvironment{LWR@HTML@minipage}{0{t} o 0{t} m}
9754 {%
9755 \LWR@traceinfo{minipage}%

```

Temporarily open a group, in which width and height is computed based on a virtual page size instead of the extra-large PDF page used during HTML tag generation.

The following used to be an actual \LaTeX minipage.

```

9756 \begingroup

```

Compute width, adjusted for frames:

```

9757 \setlength{\LWR@minipagewidth}{#4}%
9758 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}{%

```

Only create a new page if not yet nested:

```

9759 \LWR@orignewpage%

```

Adjust virtual page size:

```

9760 \addtolength{\LWR@minipagewidth}{3em}% room for frames
9761 \setlength{\linewidth}{6in}%
9762 \setlength{\textwidth}{6in}%
9763 \setlength{\textheight}{9in}%
9764 }{}%
9765 \LWR@traceinfo{computed width is \LWR@printlength{\LWR@minipagewidth}}%

```

Compute height:

```

9766 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
9767 \IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%

```

Track nesting depth:

```

9768 \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.

```

9769 \ifbool{FormatWP}{\newline}{}%
9770 \LWR@stoppars%

```

If FormatWP, add a text frame:

```

9771 \ifbool{FormatWP}{%
9772
9773 \addtocounter{LWR@thisautoidWP}{1}%
9774 \LWR@htmltag{%
9775     div id="\LWR@print@mbox{autoidWP-\arabic{LWR@thisautoidWP}}" %
9776     class="wpminipage"%
9777 }%
9778
9779 }{}%

```

Create the <div> tag with optional alignment style:

```

9780 \LWR@traceinfo{minipage: creating div class}%
9781 \LWR@htmltag{div class="minipage" style="%
9782 \ifthenelse{\equal{#1}{t}}{\LWR@print@mbox{vertical-align:bottom} ; }{}%
9783 \ifthenelse{\equal{#1}{c}}{\LWR@print@mbox{vertical-align:middle} ; }{}%
9784 \ifthenelse{\equal{#1}{b}}{\LWR@print@mbox{vertical-align:top} ; }{}%
9785 \ifthenelse{\equal{#3}{t}}{\LWR@print@mbox{justify-content:flex-start} ; }{}%
9786 \ifthenelse{\equal{#3}{c}}{\LWR@print@mbox{justify-content:center} ; }{}%
9787 \ifthenelse{\equal{#3}{b}}{\LWR@print@mbox{justify-content:flex-end} ; }{}%
9788 \ifthenelse{\equal{#3}{s}}{\LWR@print@mbox{justify-content:space-between} ; }{}%

```

Print the width and optional height styles:

```

9789 \LWR@traceinfo{minipage: about to print the width of \LWR@printlength{\LWR@minipagewidth}}%
9790 \ifbool{LWR@minipagefullwidth}%
9791 {\boolfalse{LWR@minipagefullwidth}}%
9792 {%
9793     \ifthenelse{\lengthtest{#4}=\linewidth}%
9794     {}%
9795     {width:\LWR@printlength{\LWR@minipagewidth} ; }%
9796 }%
9797 \LWR@traceinfo{minipage: about to print the height}%
9798 \IfValueT{#2}{height:\LWR@printlength{\LWR@minipageheight} ; }%
9799 "%

```

Finish with an empty line to start the contents on a new line.

```

9800
9801 % 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.

```

9802 \setlength{\linewidth}{#4}% the original width
9803 \setlength{\textwidth}{6in}%
9804 \setlength{\textheight}{9in}%

```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
9805 \LWR@origraggedright%
```

Set minipage footnotes:

```
9806 \def\@mpfn{mpfootnote}%
9807 \def\thempfn{\thempfootnote}\c@mpfootnote\z@%
9808 \let\@footnotetext\@mpfootnotetext%
```

Resume paragraph tag handling for the contents of the minipage:

```
9809 \LWR@startpars%
9810 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
9811
9812 === begin minipage ===
9813
9814 }{%
9815 \LWR@traceinfo{minipage: finished starting the minipage}%
9816 }% finished \minipage
9817 {% \endminipage
```

Print pending minipage footnotes:

```
9818 \LWR@printpendingmpfootnotes%
```

End the environment with closing tag:

```
9819 \ifboolexpr{bool{FormatWP} and bool{WPMarkMinipages}}{%
9820
9821 === end minipage ===
9822
9823 }{%
9824 \LWR@stoppars%
```

The following used to be an actual \LaTeX minipage.

```
9825 \endgroup%
9826
9827 \ifbool{FormatWP}{%
9828
9829 \LWR@htmlelementend{div}%
9830
9831 }{%
9832 \LWR@htmldivclassend{minipage}%
9833
9834 \addtocounter{LWR@minipagedepth}{-1}%
9835 \LWR@startpars%
9836 \ifbool{FormatWP}{\newline}{%}
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
9837 \global\booltrue{LWR@minipagethispar}%
9838 \LWR@traceinfo{LWR@minipage: done}%
9839 }
9840
9841 \LWR@formattedenv{minipage}
```

85.4 Parbox, mbox, makebox, framebox, fbox, raisebox

for HTML output:

`\parbox` [*pos*] [*height*] [*inner-pos*] {*width*} {*text*}

A parbox uses the minipage code:

```
9842 \NewDocumentCommand{\LWR@HTML@parbox}{O{t} o O{t} m +m}
9843 {
9844 \LWR@traceinfo{parbox of width #4}%
9845 \begin{minipage}[#1][#2][#3]{#4}%
9846 #5
9847 \end{minipage}%
9848 }
9849
9850 \LWR@formatted{parbox}
```

`\mbox` {*text*} Nullified for HTML.

```
9851 \newcommand*{\LWR@HTML@mbox}[1]{#1}
9852
9853 \LWR@formatted{mbox}
```

`\makebox` (<(>)posn) [*width*] [*pos*] {*text*}

```
9854 \NewDocumentCommand{\LWR@HTML@makebox}{d() o o m}{%
```

Check for the optional width:

```
9855 \IfValueTF{#2}%
9856 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```

9857    {% scope
9858    \def\LWR@align{center}%
9859    \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
9860    \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
9861    \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%

```

To print the width argument:

```

9862    \setlength{\LWR@tempwidth}{#2}%

```

inline-block allows width and text-align to be used in a .

```

9863    \InlineClass[%
9864        \LWR@print@mbx{display:inline-block} ; %
9865        \LWR@print@mbx{text-align}:\LWR@align\ ; %
9866        width:\LWR@printlength{\LWR@tempwidth}%
9867    ]%
9868    {makebox}%

9869    {#4}%
9870    }% scope
9871 }%

```

Without a width argument, the text is simply used inline:

```

9872 {#4}% no width
9873 }
9874
9875 \LWR@formatted{makebox}

```

`\framebox` [*<width>*] [*<pos>*] {*<text>*}

```

9876 \LetLtxMacro\LWR@origframebox\framebox
9877
9878 \NewDocumentCommand{\LWR@HTML@framebox}{o o m}{%
9879 \fbox{\makebox[#1][#2]{#3}}%
9880 }
9881
9882 \LWR@formatted{framebox}

```

`\LWR@forceminwidth` {*<legth>*}

Sets `\LWR@atleastonept` to be at least 1pt.

```

9883 \newlength{\LWR@atleastonept}
9884
9885 \newcommand*{\LWR@forceminwidth}[1]{%

```

```

9886 \setlength{\LWR@atleastonept}{#1}%
9887 \ifthenelse{%
9888     \lengthtest{\LWR@atleastonept>0pt}\AND%
9889     \lengthtest{\LWR@atleastonept<1pt}}%
9890 }%
9891 {\setlength{\LWR@atleastonept}{1pt}}%
9892 {}%
9893 }

```

`\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.

```

9894 \newcommand*{\LWR@blackborderpadding}{%
9895 border:\LWR@printlength{\LWR@atleastonept} solid black ; %
9896 padding:\LWR@printlength{\fboxsep}%
9897 }

```

`\fbox` `{\langle text \rangle}`

Creates a framed inline span enclosing the text.

Create a new HTML version, but don't use it until after **xcolor** may have loaded:

```

9898 \newcommand{\LWR@HTML@fbox}[1]{%
9899 \LWR@traceinfo{HTML fbox}%
9900 \LWR@forceminwidth{\fboxrule}%
9901 \InlineClass[%
9902 \LWR@blackborderpadding%
9903 ]{fbox}{#1}
9904 }

```

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`:

```

9905 \AtBeginDocument{\LWR@formatted{fbox}}

```

`\fboxBlock` `{\langle text \rangle}` Creates a framed HTML `<div>` of the text.

First, a print-mode version. This is newly defined for print mode, so it is defined inside `warpall`.

for HTML & PRINT:

```

9906 \end{warpHTML}
9907 \begin{warpall}
9908 \let\fboxBlock\fbox
9909 \end{warpall}
9910 \begin{warpHTML}

```

for HTML output: Next, an HTML version:

```

9911 \newcommand{\LWR@HTML@fboxBlock}[1]{%
9912 \LWR@forceminwidth{\fboxrule}%
9913 \begin{BlockClass}[%
9914 \LWR@blackborderpadding%
9915 ]{fboxBlock}
9916 #1
9917 \end{BlockClass}
9918 }
9919
9920 \LWR@formatted{fboxBlock}
9921
9922 \end{warpHTML}

```

Env `fminipage` [*<align>*] [*<height>*] [*<align>*] {*<width>*}

Creates a framed HTML <div> around its contents.

for HTML & PRINT: Print version:

```

9923 \begin{warpall}
9924
9925 \newsavebox{\LWR@fminipagebox}
9926
9927 \NewDocumentEnvironment{\LWR@print@fminipage}{0{t} o 0{t} m}
9928 {%

```

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:

```

9929 \IfValueTF{#3}%
9930 {\def\LWR@thisalign{#3}}
9931 {\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.

```

9932 \IfValueTF{#2}%
9933 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
9934 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%

```

Capture the contents of the environment:

```

9935 \begin{lrbox}{\LWR@fminipagebox}%

```

Nest the contents inside an inner minipage of the desired size:

```
9936 \IfValueTF{#2}%
9937 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
9938 {\minipage[#1]{#4}}%
9939 }
9940 {%
```

Close the inner minipage and the LR box with the contents:

```
9941 \endminipage%
9942 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
9943 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
9944 \endminipage%
9945 }
9946
9947 \newenvironment{fminipage}{\LWR@print@fminipage}{\endLWR@print@fminipage}
9948
9949 \end{warpall}
```

for HTML output:

```
9950 \begin{warpHTML}
9951
9952 \NewDocumentEnvironment{LWR@HTML@fminipage}{0{t} o 0{t} m}
9953 {%
9954 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
9955 \LWR@forceminwidth{\fboxrule}%
9956 \setlength{\LWR@tempwidth}{#4}%
9957 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
9958 \begin{BlockClass}%
9959 \LWR@blackborderpadding ; %
9960 \IfValueT{#2}{height:\LWR@printlength{\LWR@tempheight} ; }%
9961 width:\LWR@printlength{\LWR@tempwidth}%
9962 ]{fminipage}%
9963 }
9964 {%
9965 \end{BlockClass}%
9966 \LWR@traceinfo{fminipage done}%
9967 }
9968
9969 \LWR@formattedenv{fminipage}

9970 \end{warpHTML}
```

`\raisebox` $\{\langle raiselen \rangle\} [\langle height \rangle] [\langle depth \rangle] \{\langle text \rangle\}$


```


9971 \begin{warpHTML}
9972
9973 \NewDocumentCommand{\LWR@HTML@raisebox}{m o o m}{%
9974 #4%
9975 }
9976
9977 \LWR@formatted{raisebox}

9978 \end{warpHTML}

```

86 Direct formatting

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

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

For high-level block and inline custom CSS classes, see section 49.8.

for HTML output: 9979 `\begin{warpHTML}`

`\LWR@HTMLtextstyle` $\{\langle FormatWP style \rangle\} \{\langle class \rangle\} \{\langle text \rangle\}$

If `FormatWP`, adds an explicit style to the text span class. This is used by `LIBREOFFICE` to mark its imported text using the given style.

```

9980 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{%
9981 \ifbool{FormatWP}%
9982 {\LWR@htmlspanclass[#1]{#2}{#3}}%
9983 {\LWR@htmlspanclass{#2}{#3}}%
9984 }

```

`\emph` $\{\langle text \rangle\}$

```

9985 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
9986 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
9987 \LetLtxMacro\emph\LWR@HTMLemph

```

`\textmd` $\{\langle text \rangle\}$

```

9988 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{%
9989 \LWR@HTMLtextstyle{font-weight:normal}{textmd}{#1}%
9990 }
9991 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
9992
9993 \LetLtxMacro\textmd\LWR@HTMLtextmd

```

`\textbf` $\{\langle text \rangle\}$

```

9994 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
9995 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
9996 \LetLtxMacro\textbf\LWR@HTMLtextbf

```

`\textrm` $\{\langle text \rangle\}$

```

9997 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{%
9998 \LWR@HTMLtextstyle{font-family:serif}{textrm}{#1}%
9999 }
10000
10001 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
10002
10003 \LetLtxMacro\textrm\LWR@HTMLtextrm

```

`\textsf` $\{\langle text \rangle\}$

```

10004 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{%
10005 \LWR@HTMLtextstyle{font-family:sans}{textsf}{#1}%
10006 }
10007 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
10008 \LetLtxMacro\textsf\LWR@HTMLtextsf

```

`\texttt` $\{\langle text \rangle\}$

```

10009 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
10010 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
10011 \LetLtxMacro\texttt\LWR@HTMLtexttt

```

`\textup` $\{\langle text \rangle\}$

```

10012 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{%
10013 \LWR@HTMLtextstyle{font-variant:normal}{textup}{#1}%
10014 }
10015
10016 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
10017
10018 \LetLtxMacro\textup\LWR@HTMLtextup

```

`\textit` $\{\langle text \rangle\}$

```
10019 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
10020 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
10021 \LetLtxMacro\textit\LWR@HTMLtextit
```

`\textsc` $\{\langle text \rangle\}$

```
10022 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{%
10023 \LWR@HTMLtextstyle{font-variant:small-caps}{textsc}{#1}%
10024 }
10025
10026 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
10027
10028 \LetLtxMacro\textsc\LWR@HTMLtextsc
```

`\textsl` $\{\langle text \rangle\}$

```
10029 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{%
10030 \LWR@HTMLtextstyle{font-style:oblique}{textsl}{#1}%
10031 }
10032
10033 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
10034
10035 \LetLtxMacro\textsl\LWR@HTMLtextsl
```

`\textnormal` $\{\langle text \rangle\}$

```
10036 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
10037 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
10038 \LetLtxMacro\textnormal\LWR@HTMLtextnormal

10039 \DeclareRobustCommand{\LWR@nullrmfamily}{\rmfamily}{}
10040 \DeclareRobustCommand{\LWR@nullsfamily}{\sfamily}{}
10041 \DeclareRobustCommand{\LWR@nullttfamily}{\ttfamily}{}
10042 \DeclareRobustCommand{\LWR@nullbfseries}{\bfseries}{}
10043 \DeclareRobustCommand{\LWR@nullmdseries}{\mdseries}{}
10044 \DeclareRobustCommand{\LWR@nullupshape}{\upshape}{}
10045 \DeclareRobustCommand{\LWR@nullslshape}{\slshape}{}
10046 \DeclareRobustCommand{\LWR@nullscshape}{\scshape}{}
10047 \DeclareRobustCommand{\LWR@nullitshape}{\itshape}{}
10048 \DeclareRobustCommand{\LWR@nullem}[1]{\rmfamily\sfamily\ttfamily\bfseries\mdseries\upshape\slshape\scshape\itshape\textnormal{#1}}
10049 \DeclareRobustCommand{\LWR@nullnormalfont}{\rmfamily\sfamily\ttfamily\bfseries\mdseries\upshape\slshape\scshape\itshape\textnormal{#1}}
```

`\LWR@nullfonts` Removes formatting during filename operations.



Use only inside a group.

The following are *not* made robust, since they must be expanded to their nullified versions.

```

10050 \newcommand*{\LWR@nullfonts}{%
10051 \LetLtxMacro\emph\LWR@nullemp%
10052 \LetLtxMacro\textmd\LWR@nulltextmd%
10053 \LetLtxMacro\textbf\LWR@nulltextbf%
10054 \LetLtxMacro\textrm\LWR@nulltextrm%
10055 \LetLtxMacro\textsf\LWR@nulltextsf%
10056 \LetLtxMacro\texttt\LWR@nulltexttt%
10057 \LetLtxMacro\textup\LWR@nulltextup%
10058 \LetLtxMacro\textit\LWR@nulltextit%
10059 \LetLtxMacro\textsc\LWR@nulltextsc%
10060 \LetLtxMacro\textsl\LWR@nulltextsl%
10061 \LetLtxMacro\textnormal\LWR@nulltextnormal%
10062 \LetLtxMacro\rmfamily\LWR@nullrmfamily%
10063 \LetLtxMacro\sffamily\LWR@nullsffamily%
10064 \LetLtxMacro\ttfamily\LWR@nullttfamily%
10065 \LetLtxMacro\bfseries\LWR@nullbfseries%
10066 \LetLtxMacro\mdseries\LWR@nullmdseries%
10067 \LetLtxMacro\upshape\LWR@nullupshape%
10068 \LetLtxMacro\slshape\LWR@nullslshape%
10069 \LetLtxMacro\scshape\LWR@nullscshape%
10070 \LetLtxMacro\itshape\LWR@nullitshape%
10071 \LetLtxMacro\em\LWR@nullem%
10072 \LetLtxMacro\normalfont\LWR@nullnormalfont%

10073 \renewcommand*{\,}{-}%
10074 \renewcommand*{~}{-}%
10075 \renewcommand*{\newline}{ }%
10076 \renewcommand*{\textellipsis}{-}%

10077 \renewcommand*{\HTMLunicode}[1]{-}%
10078 \renewcommand*{\HTMLentity}[1]{-}%

Ampersand becomes “and”, which is a short word and is then removed from the
filename.

10079 \renewcommand*{\&}{and}%

10080 \renewcommand{\textsuperscript}[1]{##1}%
10081 \renewcommand{\textsubscript}[1]{##1}%

10082 \renewcommand{\underline}[1]{##1}%

10083 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
10084 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
10085 \DeclareRobustCommand{\LWR@HTMLtextstyle}[3]{##3}%

```

Nullify math macros.

```
10086 \def\(\##1\){}%
10087 \def\[##1\]{}%
10088 \RenewDocumentCommand{\LWR@subsingledollar}{s m m m}{}%
```

Use the simpler form with `\texorpdfstring`:

```
10089 \let\texorpdfstring\relax%
10090 \newcommand{\texorpdfstring}[2]{##2}%
10091 }
```

Remembers the current font family, series, and shape.

```
10092 \newcommand*{\LWR@f@family}{rm}
10093 \newcommand*{\LWR@f@series}{md}
10094 \newcommand*{\LWR@f@shape}{up}
```

`\LWR@textcurrentfont` `{\text}`

Prints the text with the current font choices.

```
10095 \newcommand*{\LWR@textcurrentfont}[1]{%
10096 \InlineClass{%
10097     text\LWR@f@family\LWR@origtilde{}}%
10098     text\LWR@f@series\LWR@origtilde{}}%
10099     text\LWR@f@shape%
10100 }%
10101 {#1}%
10102 }
```

Env `LWR@blocktextcurrentfont` Prints the contents with the current font choices.

```
10103 \newenvironment*{\LWR@blocktextcurrentfont}{%
10104 \BlockClass{%
10105     text\LWR@f@family\LWR@origtilde{}}%
10106     text\LWR@f@series\LWR@origtilde{}}%
10107     text\LWR@f@shape%
10108 }%
10109 }\endBlockClass}
```

`\mdseries`

```
10110 \renewrobustcmd*{\mdseries}{\renewcommand*{\LWR@f@series}{md}}
```

`\bfseries`

```
10111 \renewrobustcmd*{\bfseries}{\renewcommand*{\LWR@f@series}{bf}}
```

`\rmfamily`

```
10112 \renewrobustcmd*{\rmfamily}{\renewcommand*{\LWR@f@family}{rm}}
```

`\sffamily`

```
10113 \renewrobustcmd*{\sffamily}{\renewcommand*{\LWR@f@family}{sf}}
```

`\ttfamily`

```
10114 \renewrobustcmd*{\ttfamily}{\renewcommand*{\LWR@f@family}{tt}}
```

`\upshape`

```
10115 \renewrobustcmd*{\upshape}{\renewcommand*{\LWR@f@shape}{up}}
```

`\itshape`

```
10116 \renewrobustcmd*{\itshape}{\renewcommand*{\LWR@f@shape}{it}}
```

`\scshape`

```
10117 \renewrobustcmd*{\scshape}{\renewcommand*{\LWR@f@shape}{sc}}
```

`\normalfont`

```
10118 \renewrobustcmd*{\normalfont}{\rmfamily\mdseries\upshape}
```

`\sp` $\langle text \rangle$

For **siunitx**. Must work in math mode.

```
10119 \renewcommand{\sp}[1]{\text{\<sup>#1</sup>}}
```

`\sb` $\langle text \rangle$

For **siunitx**. Must work in math mode.

```
10120 \renewcommand{\sb}[1]{\text{\<sub>#1</sub>}}
```

`\textsuperscript` $\{\langle text \rangle\}$

```
10121 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\@textsuperscript` $\{\langle text \rangle\}$

```
10122 \renewcommand{\@textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript` $\{\langle text \rangle\}$

```
10123 \AtBeginDocument{
10124 \renewrobustcmd{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
10125 }
```

`\@textsubscript` $\{\langle text \rangle\}$

```
10126 \AtBeginDocument{
10127 \renewcommand{\@textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
10128 }
```

`\up` $\{\langle text \rangle\}$ Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
10129 \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.

```
10130 \AtBeginDocument{\let\fup\textsuperscript}
```

`\underline` $\{\langle text \rangle\}$

```
10131 \renewcommand{\underline}[1]{%
10132 \LWR@HTMLtextstyle%
10133 {text-decoration:underline; text-decoration-skip: auto}%
10134 {underline}{#1}%
10135 }
```

`\LWR@overline` $\{\langle text \rangle\}$

```
10136 \newcommand{\LWR@overline}[1]{%
10137 \LWR@HTMLtextstyle%
10138   {text-decoration:overline; text-decoration-skip: auto}%
10139   {overline}{#1}%
10140 }
```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
10141 \newcommand*{\LWR@currenttextcolor}{black}
```

`\LWR@tempcolor` The color converted to HTML colorspace.

`\LWR@tempcolortwo`

```
10142 \newcommand*{\LWR@tempcolor}{}
10143 \newcommand*{\LWR@tempcolortwo}{}

```

`\LWR@findcurrenttextcolor` Sets `\LWR@tempcolor` to the current color.

```
10144 \newcommand*{\LWR@findcurrenttextcolor}{%
10145 \renewcommand{\LWR@tempcolor}{black}%
10146 }
```

`\LWR@textcurrentcolor` $\{\langle text \rangle\}$ Like `\textcolor` but uses the current `\color` instead.

```
10147 \NewDocumentCommand{\LWR@textcurrentcolor}{m}{%
10148   \renewcommand*{\LWR@currenttextcolor}{black}%
10149   #1%
10150 }

10151 \end{warpHTML}
```

87 Skips, spaces, font sizes

for HTML output: 10152 `\begin{warpHTML}`

`\,` must be redefined after `\RequirePackage{printlen}`

Direct-formatting space commands become HTML entities:

```
10153 \renewrobustcmd*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
10154 \renewrobustcmd*{\thinspace}{\HTMLUnicode{202f}} % HTML thin non-breakable space
```

```

10155
10156 \renewrobustcmd*{~}{\HTMLentity{nbsp}}
10157
10158 \renewrobustcmd*{\textellipsis}{\HTMLunicode{2026}}

```

Direct-formatting font sizes are ignored:

```

10159 \renewrobustcmd*{\normalsize}{}
10160 \renewrobustcmd*{\small}{}
10161 \renewrobustcmd*{\footnotesize}{}
10162 \renewrobustcmd*{\scriptsize}{}
10163 \renewrobustcmd*{\tiny}{}
10164 \renewrobustcmd*{\large}{}
10165 \renewrobustcmd*{\Large}{}
10166 \renewrobustcmd*{\LARGE}{}
10167 \renewrobustcmd*{\huge}{}
10168 \renewrobustcmd*{\Huge}{}

10169 \DeclareDocumentCommand{\onecolumn}{}{}
10170
10171 \DeclareDocumentCommand{\twocolumn}{0{}}{
10172
10173 #1
10174
10175 }

```

`\hfill`

```

10176 \renewcommand*{\hfill}{\quad}

```

`\hrulefill`

```

10177 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}

```

`\dotfill`

```

10178 \renewcommand*{\dotfill}{\dots}

```

`\newpage`

```

10179 \renewcommand*{\newpage}{
10180
10181 }

```

`\newline` Uses the HTML `
` element.

```
10182 \newrobustcmd*{\LWR@newlinebr}{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
10183 \LetLtxMacro\newline\LWR@newlinebr
```

`\\` Redefined to `\LWR@endoffline` or `\LWR@tabularendoffline`.

`\LWR@endoffline` * [*len*]

`\\` is assigned to `\LWR@endoffline` at `\LWR@LwarpStart`.

Inside tabular, `\\` is temporarily changed to `\LWR@tabularendoffline`.

```
10184 \LetLtxMacro\LWR@origendoffline\\
10185 \NewDocumentCommand{\LWR@endoffline}{s o}
10186 {%
10187 \newline%
10188 }
```

`\LWR@minipagestartpars` Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `<div>` to be inside a `p`, paragraphs must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, `lwarp` correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. `lwarp` tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qquad` until the end of the paragraph, when the closing `p` tag is created.

`\hspace`
`\enskip`
`\quad`
`\qquad`

When a minipage is seen, the boolean `LWR@minipagethispar` is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. `LWR@minipagethispar` is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
10189 \newcommand*{\LWR@minipagestartpars}{%
10190 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
10191 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
10192 \newcommand*{\LWR@minipagestoppars}{%
10193 \ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
10194 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
10195 \renewcommand*{\quad}{%
10196 \LWR@minipagestoppars%
10197 \HTMLUnicode{2001}%
10198 \LWR@minipagestartpars%
10199 }
```

`\qqquad` Handles special minipage & horizontal space interactions.

```
10200 \renewcommand*{\qqquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
10201 \renewcommand*{\enskip}{%
10202 \LWR@minipagestoppars%
10203 \HTMLUnicode{2000}%
10204 \LWR@minipagestartpars%
10205 }
```

Len `\LWR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:

Len `\LWR@tempheight`

Len `\LWR@tempraise`

```
10206 \newlength{\LWR@tempwidth}
10207 \newlength{\LWR@tempheight}
10208 \newlength{\LWR@tempraise}
```

`\LWR@select@html@hspace` * $\langle length \rangle$ * $\langle length \rangle$

`\hspace`

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

`\hspace{\fill}` is converted to `\hspace{2em}`, equal to `\qqquad`.

```
10209 \newcommand{\LWR@select@html@hspace}{%
10210 \RenewDocumentCommand{\hspace}{s m}{%
10211 \setlength{\LWR@tempwidth}{##2}%
```

If `\fill`, change to `\qqquad`:

```
10212 \ifnum\gluestretchorder\LWR@tempwidth>0%
10213 \setlength{\LWR@tempwidth}{2em}%
10214 \fi%
```

Only if the width is not zero:

```
10215 \ifdimcomp{\LWR@tempwidth}{=}{0pt}{\}{%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
10216    \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
10217    \ifdimcomp{\LWR@tempwidth}{=}{.16667em}%
10218    {%
10219        \HTMLUnicode{2009}% thin breakable space
10220    }%
```

Print the span with the converted width. Not rounded.

```
10221    {%
10222        \LWR@htmltagc{%
10223            span style="width:\LWR@printlength{\LWR@tempwidth}; % extra space
10224                display:inline-block"%
10225        }%
```

If formatting for a word processor, approximate with a number of \quads, in case a span of a given width is not supported:

```
10226        \ifbool{FormatWP}{%
10227            \setlength{\LWR@templengthone}{\LWR@tempwidth}%
10228            \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
10229                \quad%
10230                \addtolength{\LWR@templengthone}{-1em}%
10231            }%
10232        }{}%
```

Close the span:

```
10233        \LWR@htmltagc{/span}%
10234    }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
10235    \LWR@minipagestartpars%
10236}% width not 0
10237}%
10238 }
```

```
\LWR@select@html@nohspace * {\length}
\hspace
```

Used to disable \hspace while creating description \items.

```

10239 \newcommand{\LWR@select@html@nohspace}{%
10240     \RenewDocumentCommand{\hspace}{s m}{}%
10241 }

```

\LWR@select@print@hspace

```

10242 \newcommand*{\LWR@select@print@hspace}{%
10243     \renewrobustcmd\hspace{\@ifstar\@hspacer\@hspace}%
10244 }

```

\hspace * {*\length*}

Handles special minipage & horizontal space interactions.

```

10245 \LWR@select@html@hspace

```

\LWR@vspace * {*\length*} Nullified vspace.

```

10246 \NewDocumentCommand{\LWR@HTML@vspace}{s m}{}
10247
10248 \LWR@formatted{vspace}

```

\linebreak [*\num*] Inserts an HTML br tag.

```

10249 \renewcommand*{\linebreak}[1] [] {\newline}

```

\nolinebreak [*\num*]

```

10250 \renewcommand*{\nolinebreak}[1] [] {}

```

\pagebreak [*\num*] Starts a new paragraph.

```

10251 \renewcommand*{\pagebreak}[1] [] {
10252
10253 }

```

\nepagebreak [*\num*]

```

10254 \renewcommand*{\nepagebreak}[1] [] {}

```

\enlargethispage * {*\len*}

```

10255 \RenewDocumentCommand{\enlargethispage}{s m}{}

```

```

\clearpage
\cleardoublepage
10256 \renewcommand*{\clearpage}{}
10257 \renewcommand*{\cleardoublepage}{}

```

```
\rule  [<raise>] {<width>} {<height>}
```

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
10258 \newcommand*{\LWR@HTML@rule}[3] [] {%
```

The width is copied into a temporary \TeX length, from which comparisons and conversions may be made:

```
10259 \setlength{\LWR@tempwidth}{#2}%
```

If it's zero-width then skip the entire rule:

```

10260 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}{
10261 {}}% zero- width
10262 {}% non-zero width

```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```

10263 \ifthenelse{%
10264 \lengthtest{\LWR@tempwidth>0pt}\AND%
10265 \lengthtest{\LWR@tempwidth<1pt}}%
10266 }%
10267 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```

10268 \setlength{\LWR@tempheight}{#3}%
10269 \ifthenelse{%
10270 \lengthtest{\LWR@tempheight>0pt}\AND%
10271 \lengthtest{\LWR@tempheight<1pt}}%
10272 }%
10273 {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
10274 \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.

```
10275 \LWR@findcurrenttextcolor%
10276 \LWR@htmltagc{%
10277 span
10278 style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
10279 \ifbool{FormatWP}{\background:\LWR@currenttextcolor ; }%
```

The width and height are printed, converted to PT:

```
10280 width:\LWR@printlength{\LWR@tempwidth} ; %
10281 height:\LWR@printlength{\LWR@tempheight} ; %
```

The raise height is converted to a css transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a \LaTeX length allows a typical \LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of \LaTeX ex height, then print that to HTML with an ex unit.

```
10282 \ifblank{#1}%
10283 {}%
10284 {%
10285 \setlength{\LWR@tempraise}{Opt-#1}%
10286 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
10287 \LWR@orignewline%
10288 -ms-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
10289 \LWR@orignewline%
10290 -webkit-transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
10291 \LWR@orignewline%
10292 transform: translate(0pt,\LWR@printlength{\LWR@tempraise}); %
10293 \LWR@orignewline%
10294 }%
```

Display inline-block to place the span inline with the text:

```
10295 display:inline-block;"%
10296 }%
```

If formatting for a word processor, approximate with a number of underscores, in case a span of a given width is not supported:

```
10297 \ifbool{FormatWP}{%
```

```

10298          \setlength{\LWR@templengthone}{\LWR@tempwidth}%
10299          \whiledo{\lengthtest{\LWR@templengthone>1em}}{%
10300              \_{}%
10301              \addtolength{\LWR@templengthone}{-1em}%
10302          }%
10303      }{}%
```

Close the span:

```
10304      \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```

10305      \LWR@minipagestartpars%
10306 }% non-zero width
10307 }
10308
10309 \LWR@formatted{rule}

10310 \end{warpHTML}
```

88 \phantomsection

for HTML output: 10311 \begin{warpHTML}

\phantomsection Emulate the **hyperref** \phantomsection command, often used to insert the bibliography into the table of contents. Ignores \ForceHTMLTOC.

```

10312 \DeclareDocumentCommand{\phantomsection}{}{}%
10313 \begingroup%
10314 \boolfalse{LWR@forcinghtmltoc}%
10315 \section*{}%
10316 \endgroup%
10317 }

10318 \end{warpHTML}
```

89 \LaTeX and other logos

Logos for HTML and print modes:

Some of these logos may be redefined in a later package, so after loading other packages, and at the beginning of the document, their definitions are finally `\let` in `\LWR@LwarpStart`.

For css conversions, see:

<http://edward.oconnor.cx/2007/08/tex-poshlet>

<http://nitens.org/taraborelli/texlogo>

89.1 HTML logos

for HTML output: 10319 `\begin{warpHTML}`

`\TeX` $\mathrm{T}_{\mathrm{E}}\mathrm{X}$

`latexlogo` is a css class used to properly typeset the E and A in $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ and friends.

`latexlogofont` is a css class used to select the font for the rest of the logo in $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{LuaT}_{\mathrm{E}}\mathrm{X}$, $\mathrm{ConT}_{\mathrm{E}}\mathrm{Xt}$, etc.

```
10320 \let\LWR@origTeX\TeX
10321
10322 \newcommand*{\LWR@TeX}
10323 {%
10324     \InlineClass{latexlogofont}%
10325     {%
10326         \LWR@HTMLtextstyle%
10327         {text-transform:uppercase}%
10328         {latexlogo}%
10329         {T\textsubscript{e}X}%
10330     }%
10331 }
```

`\LaTeX` $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, $\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$

`\LaTeXe`

```
10332 \let\LWR@origLaTeX\LaTeX
10333
10334 \newcommand*{\LWR@LaTeX}
10335 {%
10336     \InlineClass{latexlogofont}%
10337     {%
10338         \LWR@HTMLtextstyle%
10339         {text-transform:uppercase}%
10340         {latexlogo}%
10341         {L\textsuperscript{a}T\textsubscript{e}X}%
10342     }%
10343 }
```

```

10344
10345 \let\LWR@origLaTeXe\LaTeXe
10346
10347 \renewcommand*{\LaTeXe}
10348 {\LaTeX\InlineClass{latexlogofont}%
10349 {\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}

```

`\LuaTeX` `LuaTEX`, `LuaWTEX`

`\LuaLaTeX`

```

10350 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
10351 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}

```

`\XeTeX` `XETEX`, `XEWTEX`

`\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 `XEWTEX` and spaces `WETEX` appropriately.

```

10352 \newcommand*{\Xe}
10353 {X\textsubscript{\HTMLUnicode{18e}}}
10354 \newcommand*{\LWR@XeTeX}{\InlineClass{xetexlogo}{\Xe}\TeX}
10355 \newcommand*{\LWR@XeLaTeX}{\InlineClass{xelatexlogo}{\Xe}\LaTeX}

```

`\ConTeXt` `ConTEXt`

```

10356 \newcommand*{\LWR@ConTeXt}
10357 {\InlineClass{latexlogofont}{Con}\TeX{}}%
10358 \InlineClass{latexlogofont}{t}}

```

`\BibTeX` `BIBTEX`, *MakeIndex*

`\MakeIndex`

```

10359 \providecommand*{\BibTeX}
10360 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
10361
10362 \newcommand*{\MakeIndex}
10363 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

`\AmS` \mathcal{MS}

`amslogo` is a css class used for the \mathcal{MS} logo.

```

10364 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{-}
10365 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}

```

`\MiKTeX` `MiKTeX`

```
10366 \newcommand*{\MiKTeX}{\InlineClass{latexlogofont}{MiK}\TeX}
```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the LyX logo.

```
10367 \newcommand*{\LyX}{\InlineClass{lyxlogo}{LyX}}
```

```
10368 \end{warpHTML}
```

89.2 Print logos

for PRINT output:

```
10369 \begin{warpprint}
10370 \newcommand*{\XeTeXrevE}
10371   {\hspace{- .1667em}\raisebox{- .5ex}{\reflectbox{E}}\hspace{- .125em}}
10372 \providecommand*{\XeTeX}{\mbox{X\XeTeXrevE\TeX}}
10373 \providecommand*{\XeLaTeX}{\mbox{X\XeTeXrevE\LaTeX}}
10374 \providecommand*{\AmS}{%
10375   \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
10376   \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
10377 \newcommand*{\LyX}{\textsf{LyX}}
10378 \providecommand*{\LuaTeX}{\mbox{Lua\TeX}}
10379 \providecommand*{\LuaLaTeX}{\mbox{Lua\LaTeX}}
10380 \providecommand*{\BibTeX}{\mbox{B\textsc{ib}\TeX}}
10381 \providecommand*{\MakeIndex}{\mbox{\textit{MakeIndex}}}
10382 \providecommand*{\ConTeXt}{\mbox{Con\TeXt}}
10383 \providecommand*{\MiKTeX}{\mbox{MiK\TeX}}
10384 \end{warpprint}
```

90 `\AtBeginDocument`, `\AtEndDocument`

for HTML output:

```
10385 \begin{warpHTML}
```

`\LWR@LwarpStart` Automatically sets up the HTML-related actions for the start and end of the document.

`\LWR@LwarpEnd`

```
10386 \AfterEndPreamble{\LWR@LwarpStart}
```

```
10387 \AtEndDocument{\LWR@LwarpEnd}
```

```
10388 \end{warpHTML}
```

91 Koma-script

Load patches to **koma-script**.

for HTML output: 10389 \begin{warpHTML}

```
10390 \@ifclassloaded{scrbook}{\RequirePackage{lwarp-patch-komascript}}{}
10391 \@ifclassloaded{scrartcl}{\RequirePackage{lwarp-patch-komascript}}{}
10392 \@ifclassloaded{scrreprt}{\RequirePackage{lwarp-patch-komascript}}{}

10393 \end{warpHTML}
```

92 Memoir

Load patches to **memoir**.

for HTML output: 10394 \begin{warpHTML}

```
10395 \@ifclassloaded{memoir}{\RequirePackage{lwarp-patch-memoir}}{}

10396 \end{warpHTML}
```

93 Trademarks

- \TeX is a trademark of American Mathematical Society.
- ADOBE® and ADOBE **Framemaker**® are either registered trademarks or trademarks of ADOBE SYSTEMS INCORPORATED in the United States and/or other countries.
- LINUX® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- MAC OS® is a trademark of APPLE INC.
- MADCAP FLARE™ is the property of MADCAP SOFTWARE, INC.
- MATHJAX is copyright 2009 and later. The MATHJAX CONSORTIUM is a joint venture of the AMERICAN MATHEMATICAL SOCIETY (AMS) and the SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS (SIAM) to advance mathematical and scientific content on the web.
- MICROSOFT®, ENCARTA, MSN, and WINDOWS® are either registered trademarks or trademarks of MICROSOFT CORPORATION in the United States and/or other countries.
- UNIX® is a registered trademark of THE OPEN GROUP.

The following adjustments apply to the lwarp-* package listings:

File 2 **lwarp-a4.sty**

§ 94 Package **a4**

Pkg a4 **a4** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4}
 2 \newcommand*{\WideMargins}{}

File 3 **lwarp-a4wide.sty**

§ 95 Package **a4wide**

Pkg a4wide **a4wide** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4wide}

File 4 **lwarp-a5comb.sty**

§ 96 Package **a5comb**

Pkg a5comb **a5comb** is ignored.


for HTML output: 1 \LWR@ProvidesPackageDrop{a5comb}

File 5 **lwarp-abstract.sty**

§ 97 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*{\abstitleskip}%
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{\abstitleskip}%
```

```

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{\abstitlekip}%
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

§ 98 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**

§ 99 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**

§ 100 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@print@mbx{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
14 \LWR@print@mbx{margin-right:\LWR@printlength{\LWR@templengthtwo}}%
15 ]{\LWR@mccolstype}%
16 }
17 {\endBlockClass}

```

File 9 **lwarp-addlines.sty**

§ 101 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-ae.sty**

§ 102 Package **ae**

Pkg ae **ae** does not work with **pdftotext**, and is superceded by **latinmodern**.

for HTML output: 1 \LWR@loadnever{ae}{latinmodern}

File 11 **lwarp-aecc.sty**

§ 103 Package **aecc**

Pkg aecc **aecc** does not work with **pdftotext**, and is superceded by **latinmodern**.

for HTML output: 1 \LWR@loadnever{aecc}{latinmodern}

File 12 **lwarp-afterpage.sty**

§ 104 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 13 **lwarp-algorithm2e.sty**

§ 105 Package **algorithm2e**

(Emulates or patches code by CHRISTOPHE FIORIO.)

Pkg **algorithm2e** **algorithm2e** is patched for use by **lwarp**.

For print output, captions are placed according to package options, but for HTML output captions are placed where used. Therefore, to have captions appear at the top of the algorithms for both print and HTML, place each captions at the top of each algorithm.

for HTML output: 1 \LWR@ProvidesPackagePass{algorithm2e}

For the list-of entries:

```
2 \renewcommand{\l@algocf}[2]{\hypertocfloat{1}{algocf}{loa}{#1}{#2}}
```

Select the **lwarp** float style according to the **algorithm2e** style:

```
3 \newcommand*{\LWR@floatstyle@algocf}{ruled}
4
5 \ifdefstring{\algocf@style}{boxed}{%
6 \renewcommand*{\LWR@floatstyle@algocf}{boxed}
7 }{}
8
9 \ifdefstring{\algocf@style}{boxruled}{%
```

```

10 \renewcommand*{\LWR@floatstyle@algocf}{boxruled}
11 }{}
12
13 \ifdefstring{\algocf@style}{plain}{%
14 \renewcommand*{\LWR@floatstyle@algocf}{plain}
15 }{}

```

Paragraph handling to allow line numbers under certain conditions:

```

16 \newbool{LWR@algocf@dopars}
17 \booltrue{LWR@algocf@dopars}
18
19 \renewcommand{\algocf@everypar}{%
20 \ifbool{LWR@algocf@dopars}{%
21   \ifbool{LWR@doingstartpars}{%
22     \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}{%
23       }{}%
24     }{}%
25     \algocf@everypar\nl\algocf@everyparhanging%
26   }{}%
27 }{}%
28 }{}%
29 }

```

lwarp caption handling:

```

30 \renewcommand{\algocf@makecaption}[2]{%
31 \LWR@caption@begin{algocf}%
32 \algocf@captiontext{#1}{#2}%
33 \LWR@caption@end%
34 }

```

Print any caption where it is declared:

```

35 \renewcommand{\algocf@makecaption@plain}[2]{%
36   \LWR@caption@begin{algocf}%
37   \algocf@captiontext{#1}{#2}%
38   \LWR@caption@end%
39 }
40
41 \renewcommand{\algocf@makecaption@boxed}[2]{%
42   \LWR@caption@begin{algocf}%
43   \algocf@captiontext{#1}{#2}%
44   \LWR@caption@end%
45 }
46
47 \renewcommand{\algocf@makecaption@ruled}[2]{%
48   \LWR@caption@begin{algocf}%
49   \algocf@captiontext{#1}{#2}%

```

```

50   \LWR@caption@end%
51 }

```

Turn off line numbering while making the caption:

```

52 \long\def\algocf@latexcaption#1[#2]#3{% original definition of caption
53 \boolfalse{LWR@algocf@dopars}% lwarp
54   \par%
55   \addcontentsline{\csname ext@#1\endcsname}{#1}%
56   {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
57   \begingroup%
58   \@parboxrestore%
59   \if@minipage%
60     \setminipage%
61     \fi%
62     \normalsize%
63     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par%
64     \endgroup%
65 \booltrue{LWR@algocf@dopars}% lwarp
66 }

```

Line numbers are printed in a of class alg2elinenumber:

```

67 \renewcommand{\algocf@printnl}[1]{%
68   \InlineClass{alg2elinenumber}{\NlSty{#1}}~%
69 }%

```

While initializing an algorithm environment, locally declare the style of a regular figure to be the same as the algorithm style, in case the figure option was used.

```

70 \preto{\algocf@init}{%
71   \edef\LWR@floatstyle@figure{\LWR@floatstyle@algocf}%
72 }

```

For **lwarp**, the algorithm is not assembled inside a box, since lateximages would not work, so the captions are printed where declared.

```

73 \renewcommand{\@algocf@start}{%
74   \let\@mathsemicolon=\;;\def\;;{\ifmmode\@mathsemicolon\else\@endalgoln\fi}%
75   \raggedright%
76   \AlFnt{}%
77   \booltrue{LWR@algocf@dopars}% lwarp
78 }
79
80 \renewcommand{\@algocf@finish}{%
81   \boolfalse{LWR@algocf@dopars}% lwarp
82   \lineskip\normallineskip\setlength{\skiptotal}{\@defaultskiptotal}%
83   \let\;=\@mathsemicolon%
84   \let\]=\@mathdisplay%

```

```
85 }
```

Use an HTML break:

```
86 \renewcommand{\BlankLine}{%
87 \LWR@stoppars%
88 \LWR@htmltagc{br /}%
89 \LWR@startpars%
90 }
```

Simplified for HTML. The paragraph handling must be preserved.

```
91 \renewcommand{\SetKwInOut}[2]{%
92   \algocf@newcommand{#1}[1]{%
93     \ifthenelse{\boolean{algocf@hanginginout}}{%
94       {\relax}%
95       {\algocf@seteveryparhanging{\relax}}%
96     \ifthenelse{\boolean{algocf@inoutnumbered}}{%
97       {\relax}%
98       {\algocf@seteveryparnl{\relax}}%
99     }%
100       \KwSty{#2\algocf@typo:}%
101       ~##1\par%
102     }%
103     \algocf@linesnumbered% reset the numbering of the lines
104     \ifthenelse{\boolean{algocf@hanginginout}}{%
105       {\relax}%
106       {\algocf@reseteveryparhanging}%
107     }%
108   }%
109 }
110 \renewcommand{\ResetInOut}[1]{}%

```

Each of the following creates a <div> of a given class, and turns off line numbering while creating the <div> tags:

```
111 \renewcommand{\algocf@Vline}[1]{%
112 \boolfalse{LWR@algocf@dopars}%
113 \begin{BlockClass}{alg2evline}
114 \booltrue{LWR@algocf@dopars}%
115 #1
116 \boolfalse{LWR@algocf@dopars}%
117 \end{BlockClass}
118 \booltrue{LWR@algocf@dopars}%
119 }

120 \renewcommand{\algocf@Vsline}[1]{%
121 \boolfalse{LWR@algocf@dopars}%
122 \begin{BlockClass}{alg2evsline}

```

```

123 \booltrue{LWR@algocf@dopars}%
124 #1
125 \boolfalse{LWR@algocf@dopars}%
126 \end{BlockClass}
127 \booltrue{LWR@algocf@dopars}%
128 }

129 \renewcommand{\algocf@Noline}[1]{%
130 \boolfalse{LWR@algocf@dopars}%
131 \begin{BlockClass}{alg2enoline}
132 \booltrue{LWR@algocf@dopars}%
133 #1
134 \boolfalse{LWR@algocf@dopars}%
135 \end{BlockClass}
136 \booltrue{LWR@algocf@dopars}%
137 }

```

The [H] environment is converted to a regular float, which in HTML is placed where declared. Reusing the regular float allows the [H] version to reuse the ruled and boxed options.

```

138 \LetLtxMacro\algocf@Here\algocf
139 \LetLtxMacro\endalgocf@Here\endalgocf

```

File 14 **lwarp-algorithmicx.sty**

§ 106 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 354.1.

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 \whiledo{\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 15 **lwarp-alltt.sty**

§ 107 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{3}{alltt}%
7 }
8 \AfterEndEnvironment{alltt}{%
9   \LWR@afterendverbatim{2}%
10 }
11 }

```

File 16 **lwarp-amsthm.sty**

§ 108 Package **amsthm**

(Emulates or patches code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

The original source code is located in `amscldx.dtx`, and printed in `amscldx.pdf`.

Pkg `amsthm` **amsthm** is patched for use by **lwarp**.

Table 12: AMStm package — CSS styling of theorems and proofs

Theorem: `<div>` of class `amsthmbdy<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\exp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\exp\@nx\csname th@\the\thm@style\endcsname}%
22           }\{##1}\}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\@ynthm{#2}}[]}%
25       \fi
26       \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#2}}}% 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}{\csname 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 \ifundefined{LWR@orig@openbox}{
111 \LetLtxMacro\LWR@orig@openbox\openbox
112 \LetLtxMacro\LWR@orig@blacksquare\blacksquare
113 \LetLtxMacro\LWR@orig@Box\Box
114
115 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
116 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
117 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
118
119 \appto\LWR@restoreorigformatting{%
120 \LetLtxMacro\openbox\LWR@orig@openbox%
121 \LetLtxMacro\blacksquare\LWR@orig@blacksquare%
122 \LetLtxMacro\Box\LWR@orig@Box%
123 }% \appto
124 }{}% \ifundefined
125 }% AtBeginDocument

```

Patched for CSS:

```

126 \renewenvironment{proof}[1][\proofname]{\par
127 \LWR@forcenewpage% lwarp
128 \BlockClass{amsthmproof}% lwarp
129 \pushQED{\qed}%
130 \normalfont \topsep6\p@\@plus6\p@\relax
131 \trivlist
132 \item[
133 \InlineClass{amsthmproofname}{#1\@addpunct{.}}\ignorespaces% changes
134 ]{}%
135 \InlineClass{theoremendmark}{\popQED}\endtrivlist%
136 \endBlockClass% lwarp
137 \@endpefalse
138 }

```

File 17 **lwarp-anonchap.sty**

§ 109 Package **anonchap**

(Emulates or patches code by PETER WILSON.)

Pkg anonchap **anonchap** is emulated.



tocloft & other packages

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

The code is shared by **tocbibind**.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{anonchap}

2 \newcommand{\simplechapter}[1][\@empty]{%
3   \def\@chapcntformat##1{%
4     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
5   }%
6 }
7
8 \newcommand{\restorechapter}{%
9 \let\@chapcntformat\@seccntformat%
10 }
```

File 18 **lwarp-anysize.sty**

§ 110 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 19 **lwarp-appendix.sty**

§ 111 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 20 **lwarp-arabicfront.sty**

§ 112 Package **arabicfront**

Pkg arabicfront **arabicfront** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{arabicfront}
```

File 21 **lwarp-array.sty**

§ 113 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 22 **lwarp-arydshln.sty**

§ 114 Package **arydshln**

(Emulates or patches code by HIROSHI NAKASHIMA.)

Pkg arydshln **arydshln** heavily patches tabular code, so the actual package is not used. **arydshln** is emulated for HTML tabular, and reverts to solid rules for SVG math array and tabular in a lateximage.

CSS is not able to display a double-dashed border, so a single-dashed rule is displayed as a single-dashed border, and a double-dashed rule is displayed as a thicker single-dashed border.

for HTML output: **array** is required to allow `\newcolumn` below.

```

1 \RequirePackage{array}

2 \LWR@ProvidesPackageDrop{arydshln}

Ignored, but included for source compatibility:

3 \newdimen\dashlinedash \dashlinedash4pt %

```

```

4 \newdimen\dashlinegap \dashlinegap4pt %
5 \let\hdashlinewidth\dashlinedash
6 \let\hdashlinegap\dashlinegap
7
8 \def\ADLnullwide{}
9 \def\ADLsomewide{}
10 \def\ADLnullwidehline{}
11 \def\ADLsomewidehline{}
12
13 \def\ADLactivate{}
14 \def\ADLinactivate{}
15 \newcommand*{\ADLdrawingmode}[1]{}
16 \newcommand*{\ADLnoshorthanded}{}
17 \newcommand*{\dashgapcolor}[2][{}]{
18 \newcommand*{\nodashgapcolor}{}

```

In a `lateximage`, revert to solid vertical rules:

```

19 \appto\LWR@restoreorigformatting{
20 \newcolumnntype{:}{|}%
21 \newcolumnntype{;}[1]{|}%
22 \LetLtxMacro\hdashline\hline%
23 }

```

Some of these macros are already defined as temporary placeholders in the **lwarp** core, so they must be redefined here.

The emulated defaults also work for an emulated print mode inside a `lateximage`:

```

24 \def\hdashline{
25 % \adl@hdashline\adl@ihdashline
26 % \adl@hdashline\adl@inactivehdl
27 }
28 \def\adl@hdashline#1{\noalign{\ifnum0='}\fi
29 % \ifadl@zwhrule \vskip-\arrayrulewidth
30 % \else
31 % \adl@hline\adl@connect\arrayrulewidth
32 % \hrule \@height \arrayrulewidth% lwarp
33 % \fi
34 % \@ifnextchar[%]
35 % {\#1}%
36 % {\#1[%
37 % \dashlinedash/\dashlinegap
38 % 1pt/1pt
39 % ]}}
40 % \def\adl@ihdashline[#1/#2]{\ifnum0='{\fi}%
41 % \multispan{\adl@columns}\unskip \adl@hcline\z@[#1/#2]%
42 % \noalign{\ifnum0='}\fi
43 % \futurelet\@tempa\adl@xhline}

```

```

44 \def\adl@inactivehdl[#1/#2]{
45 %      \ifadl@zwhrule \vskip-\arrayrulewidth \fi
46 %      \hrule\@height\arrayrulewidth
47 %      \futurelet\@tempa\adl@xhline}
48 \def\adl@xhline{\ifx\@tempa\hline \adl@ixhline\fi
49 %      \ifx\@tempa\hdashline \adl@ixhline\fi
50 %      \ifnum0='{ \fi}}
51 \def\adl@ixhline{\vskip\doublerulesep \adl@hline\relax\doublerulesep}
52 \def\adl@hline#1#2{%
53 % \@tempcnta#2
54 %      \global\advance\adl@totalheight\@tempcnta
55 %      \xdef\adl@rowsL{\adl@rowsL
56 %          (#1/\number\@tempcnta);}
57 %      \xdef\adl@rowsR{\adl@rowsR
58 %          (#1/\number\@tempcnta);}
59 }
60
61 \def\cdashline#1{\noalign{\ifnum0='{ \fi
62 %      \@ifnextchar [%]
63 %          {\adl@cdline[#1]]}%
64 %          {\adl@cdline[#1][\dashlinedash/\dashlinegap]}
65 %          {\adl@inactivecdl[#1]]}%
66 %          {\adl@inactivecdl[#1][\dashlinedash/\dashlinegap]}
67 }
68
69 \def\adl@inactivecdl[#1-#2][#3]{\ifnum0='{ \fi}\cline{#1-#2}}

```

File 23 **lwarp-atbegshi.sty**

§ 115 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 24 **lwarp-authblk.sty**

§ 116 Package **authblk**

(Emulates or patches code by PATRICK W. DALY.)

Pkg **authblk** **authblk** is patched for HTML.

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

 **load order**

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

(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 \LWRAB@ProvidesPackagePass{authblk}

```

Patch to add a class for the affiliation:

```

8 \LetLtxMacro\LWRAB@affil\affil
9
10 \renewcommand{\affil}[2][]{%
11 \LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
12 }

```

Create an HTML break for an **\authorcr**:

```
13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

File 25 **lwarp-axodraw2.sty**

§ 117 Package **axodraw2**

(Emulates or patches code by JOHN C. COLLINS, J.A.M. VERMASEREN.)

Pkg axodraw2 **axodraw2** is patched for use by **lwarp**.

for HTML output:

```
1 \LWR@ProvidesPackagePass{axodraw2}


2 \BeforeBeginEnvironment{axopicture}{\begin{lateximage}[(-axopicture--\packagediagramname)]}
3
4 \AfterEndEnvironment{axopicture}{\end{lateximage}}
```

File 26 **lwarp-backref.sty**

§ 118 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 27 **lwarp-balance.sty**

§ 119 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 28 **lwarp-bibunits.sty**

§ 120 Package **bibunits**

(Emulates or patches code by THORSTEN HANSEN.)

Pkg bibunits **bibunits** is patched for use by **lwarp**.

for HTML output:

```
1 \LWR@ProvidesPackagePass{bibunits}

2 \def\bu@bibdata{\BaseJobname}
```

File 29 **lwarp-bigdelim.sty**

§ 121 Package **bigdelim**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg bigdelim **bigdelim** is used as-is for print or lateximage, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the `\multirow` which indicates the actual height of the delimiter. The delimiter character is given a `` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`

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

```

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

```

```

<>      a      b

left {  c      d
      e      f

<>      g      h

```

for HTML output:

First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case **bigdelim** was not loaded:

```

1 \let\ldelim\relax
2 \let\rdelim\relax

```

Next, load the package's new definitions:

```

3 \LWR@ProvidesPackagePass{bigdelim}

\ldelim {\langle 1:delimiter \rangle} {\langle 2:#rows \rangle} {\langle 3:width \rangle} [\langle 4:text \rangle]
\rdelim

4 \NewDocumentCommand{\LWR@HTML@ldelim}{m m m O{}}{%
5 \renewcommand{\LWR@multirowborder}{right}%
6 \multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
7 }
8
9 \LWR@formatted{ldelim}
10
11 \NewDocumentCommand{\LWR@HTML@rdelim}{m m m O{}}{%
12 \renewcommand{\LWR@multirowborder}{left}%
13 \multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
14 }
15
16 \LWR@formatted{rdelim}

```

File 30 **lwarp-bigstrut.sty**

§ 122 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 31 **lwarp-blowup.sty**

§ 123 Package **blowup**

Pkg blowup **blowup** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{blowup}

2 \newcommand*\blowUp[1]{}

```

File 32 **lwarp-bookmark.sty**

§ 124 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 33 **lwarp-booktabs.sty**

§ 125 Package **booktabs**

(Emulates or patches code by SIMON FEAR.)

Pkg booktabs **booktabs** is emulated during HTML output, and used as-is during print output and inside an HTML lateximage.

for HTML output: First, forget the placeholder macros:

```

1 \LetLtxMacro\toprule\relax
2 \LetLtxMacro\midrule\relax
3 \LetLtxMacro\cmidrule\cline
4 \LetLtxMacro\bottomrule\relax
5 \LetLtxMacro\addlinespace\relax
6 \LetLtxMacro\morecmidrules\relax
7 \LetLtxMacro\specialrule\relax
8
9 \LWR@ProvidesPackagePass{booktabs}

10 \DeclareDocumentCommand{\LWR@HTML@toprule}{o d()}%
11   {%
12     \IfValueTF{#1}%
13       {\LWR@docmidrule[#1] () {1-\arabic{LWR@tabletotalcols}}}%
14       {%
15         \ifbool{FormatWP}%
16           {\LWR@docmidrule[#1] () {1-\arabic{LWR@tabletotalcols}}}%
17           {\booltrue{LWR@doingtbrule}}}%
18       }%
19   \LWR@getmynexttoken}
20
21 \LWR@expandableformatted{toprule}
22
23 \DeclareDocumentCommand{\LWR@HTML@midrule}{o d()}%
24   {%
25     \IfValueTF{#1}%
26       {\LWR@docmidrule[#1] () {1-\arabic{LWR@tabletotalcols}}}%
27       {%
28         \ifbool{FormatWP}%
29           {\LWR@docmidrule[#1] () {1-\arabic{LWR@tabletotalcols}}}%
30           {\addtocounter{LWR@hlines}{1}}}%
31       }%
32   \LWR@getmynexttoken}
33
```

```

34 \LWR@expandableformatted{midrule}
35
36 \DeclareDocumentCommand{\LWR@HTML@cmidrule}{O{\LWR@cmidrulewidth} d() m}{%
37   \LWR@docmidrule[#1](#2){#3}%
38   \LWR@getmynexttoken%
39 }%
40
41 \LWR@expandableformatted{cmidrule}
42
43 \DeclareDocumentCommand{\LWR@HTML@bottomrule}{o d()}{%
44   \IfValueTF{#1}%
45     {\LWR@docmidrule[#1](){1-\arabic{LWR@tabletotalcols}}}%
46     {%
47       \ifbool{FormatWP}%
48         {\LWR@docmidrule[#1](){1-\arabic{LWR@tabletotalcols}}}%
49         {\booltrue{LWR@doingtbrule}}}%
50     }%
51   \LWR@getmynexttoken%
52 }%
53
54 \LWR@expandableformatted{bottomrule}
55
56 \DeclareDocumentCommand{\LWR@HTML@addlinespace}{o}{}%
57
58 \LWR@expandableformatted{addlinespace}
59
60 \DeclareDocumentCommand{\LWR@HTML@morecmidrules}{-}{}%
61
62 \LWR@expandableformatted{morecmidrules}
63
64 \DeclareDocumentCommand{\LWR@HTML@specialrule}{m m m d()}{%
65   {\LWR@docmidrule[#1](){1-\arabic{LWR@tabletotalcols}}\LWR@getmynexttoken}%
66
67 \LWR@expandableformatted{specialrule}

```

File 34 **lwarp-boxedminipage.sty**

§ 126 Package **boxedminipage**

Pkg boxedminipage **boxedminipage** is superseded by **boxedminipage2e**.

for HTML output: 1 \LWR@loadnever{boxedminipage}{boxedminipage2e}

File 35 **lwarp-boxedminipage2e.sty**

§ 127 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 36 **lwarp-breakurl.sty**

§ 128 Package **breakurl**

(Emulates or patches code by VILAR CAMARA NETO.)

Pkg breakurl **breakurl** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{breakurl}

```
2 \LetLtxMacro\burl?url
3
4 \NewDocumentCommand{\LWR@burlaltb}{0{} +m m}{%
5 \LWR@ensuredoingapar%
6 \LWR@subhyperref{#2}%
7 \LWR@subhyperref{#3}%
8 \endgroup% restore catcodes
9 }
10
11 \newrobustcmd*{\burlalt}{%
12 \begingroup%
13 \catcode'\#12%
```

```

14 \catcode'\%=12%
15 \catcode'\&=12%
16 \catcode'\~=12%
17 \catcode'\_ =12%
18 \LWR@burlaltb%
19 }
20
21 \LetLtxMacro\urlalt\burlalt

```

File 37 **lwarp-bytefield.sty**

§ 129 Package **bytefield**

(Emulates or patches code by SCOTT PAKIN.)

Pkg bytefield **bytefield** is patched for use by **lwarp**.

for HTML output:

```

1 \LWR@ProvidesPackagePass{bytefield}

2 \BeforeBeginEnvironment{bytefield}{\begin{lateximage}[(-bytefield-~\packagediagramname)]}
3
4 \AfterEndEnvironment{bytefield}{\end{lateximage}}

```

File 38 **lwarp-cancel.sty**

§ 130 Package **cancel**

Pkg cancel **cancel** is used as-is for SVG math, and emulated for HTML text output.

for HTML output:

```

1 \LWR@origRequirePackage{lwarp-xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackagePass{cancel}

```

`\cancelto` is math-only, so is used as-is.

```

3 \LetLtxMacro\LWR@origcancel\cancel
4 \LetLtxMacro\LWR@origbcancel\bcancel
5 \LetLtxMacro\LWR@origxcancel\xcancel
6
7 \appto\LWR@restoreorigformatting{%
8 \LetLtxMacro\cancel\LWR@origcancel%
9 \LetLtxMacro\bcancel\LWR@origbcancel%
10 \LetLtxMacro\xcancel\LWR@origxcancel%
11 }

```

```

\LWR@cancelcolor  {\text} {\color} {\class} {\colorstyle} {\FormatWPstyle}
Add colors if not empty:
12 \newcommand{\LWR@cancelcolor}[5]{%
13 \ifcsempy{#2}%
14 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
15 {\LWR@htmlspanclass{#5;#4:\LWR@origpound\LWR@tempcolor}{#3}{#1}}%
16 }

\cancel  {\text}

17 \DeclareRobustCommand{\cancel}[1]{%
18 \begingroup%
19 \CancelColor%
20 \LWR@findcurrenttextcolor%
21 \color{black}%
22 \LWR@cancelcolor{#1}{\LWR@tempcolor}{sout}{text-decoration-color}%
23   {text-decoration:line-through}%
24 \endgroup%
25 }
26
27 \LetLtxMacro\bcancel\cancel
28 \LetLtxMacro\xcancel\cancel

```

File 39 **lwarp-caption.sty**

§ 131 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@iibox{#1}{#2}{#3}}{%
4     \wd\@tempboxa%
5     \linewidth% lwarp
6   }%
7 % \LWR@traceinfo{caption@ibox: done}%
8 }

9 \long\def\caption@iibox#1#2#3[#4]{%
10  \@testopt{\caption@iibox{#1}{#2}{#3}{#4}}\captionbox@hj@default
11 }

```

```

12 \long\def\caption@iiibox#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@iiibox 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@iiibox 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 {\captiontextfont

```

```

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 40 **lwarp-caption2.sty**

§ 132 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 41 **lwarp-ccaption.sty**

§ 133 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 42 **lwarp-changebar.sty**

§ 134 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 43 **lwarp-changepage.sty**

§ 135 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{\changepage}[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 44 **lwarp-chngpage.sty**

§ 136 Package **chngpage**

(Emulates or patches code by PETER WILSON.)

Pkg chngpage **chngpage** is emulated.

for HTML output: Discard all options for **lwarp-chngpage**:

```

1 \LWR@ProvidesPackageDrop{chngpage}
2 \LWR@origRequirePackage{changepage}

```

File 45 **lwarp-chappg.sty**

§ 137 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 46 **lwarp-chapterbib.sty**

§ 138 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 47 **lwarp-chemfig.sty**

§ 139 Package **chemfig**
(Emulates or patches code by CHRISTIAN TELLECHEA.)

Pkg chemfig **chemfig** is patched for use by **lwarp**.

The images are not hashed because they depend on external settings which may be changed at any time, and are unlikely to be reused inline anyhow.

for HTML output:

```

1 \LWR@ProvidesPackagePass{chemfig}

2 \LetLtxMacro\LWR@chemfig@origchemfig\chemfig
3
4 \DeclareDocumentCommand\chemfig{s O{} O{} m}{%
5   \begin{lateximage}[(-chemfig--\packagediagramname)]%
6   \IfBooleanTF{#1}{%
7     \LWR@chemfig@origchemfig*[#2][#3]{#4}%
8   }{%
9     \LWR@chemfig@origchemfig[#2][#3]{#4}%
10  }
11  \end{lateximage}%
12 }
13
14 \LetLtxMacro\LWR@chemfig@origCF@lewis@b\CF@lewis@b
15
16 \def\CF@lewis@b#1#2{%
17 \begin{lateximage}[(-chemfig--\packagediagramname)]%
18 \LWR@chemfig@origCF@lewis@b{#1}{#2}%
19 \end{lateximage}%
20 }
21
22 \preto{\schemestart}{\begin{lateximage}[(-chemfig--\packagediagramname)]}
23 \appto{\CF@schemestop}{\end{lateximage}}
24
25 \LetLtxMacro\LWR@chemfig@origchemleft\chemleft
26
27 \def\chemleft#1#2\chemright#3{%
28 \begin{lateximage}[(-chemfig--\packagediagramname)]%
29 \LWR@chemfig@origchemleft#1#2\chemright#3%
30 \end{lateximage}%
31 }
32
33 \LetLtxMacro\LWR@chemfig@origchemup\chemup

```

```

34
35 \def\chemup#1#2\chemdown#3{%
36 \begin{lateximage}[(-chemfig--\packagediagramname)]%
37 \LWR@chemfig@origchemup#1#2\chemdown#3%
38 \end{lateximage}%
39 }

```


File 48 **lwarp-chemformula.sty**

§ 140 Package **chemformula**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg chemformula **chemformula** is patched for use by **lwarp**.

The SVG images are hashed according to contents and local options. Global options are assumed to be constant document-wide.

 **chemformula with MATHJAX** **chemformula** works best without MATHJAX. If MATHJAX is used, `\displaymathother` must be used before array, and then `\displaymathnormal` may be used after. (The **chemformula** package adapts to array, but does not know about MATHJAX, and MATHJAX does not know about **chemformula**.)

While using MATHJAX, `\displaymathother` may also be used for other forms of display and inline math which contain **chemformula** expressions.

for HTML output: 1 \LWR@ProvidesPackagePass{chemformula}[2017/03/23]

2 \ExplSyntaxOn

`\ch` Enclose in an inline SVG image or MathJax. The `alt` tag is the contents of the `\ch` expression. The filename is hashed, and also has additional hashing information based on the local options.

```

3 \RenewDocumentCommand \ch { 0{}m }
4 { %

```

To work inside align with `\displaymathother`, a simple version must be used to work with **chemformula**'s adaptation to align.

```

5    \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}%    lwarp
6    {
7       \chemformula_ch:nn {#1} {#2}%    original
8    }

```

If used as the outer level, must temporarily ensure MATHJAX is disabled:

```

9      {
10         \begingroup%
11         \boolfalse{mathjax}%

```

An inline image is used, adjusted for the baseline:

```

12         \LWR@subsingledollar*{% lwarp
13             \textbackslash{}ch\{\LWR@HTMLsanitize{#2}\}% alt text
14         }{%
15             \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}% add'l hashing
16         }%
17         {%
18             \chemformula_ch:nn {#1} {#2}% original
19         }%
20         \endgroup%
21     }
22 }

```

`\chcpd` Similar to `\ch`.

```

23 \cs_gset_protected:Npn \chemformula_chcpd:nn #1#2
24 {
25     \begingroup%
26     \boolfalse{mathjax}%
27     \LWR@subsingledollar*{% lwarp
28         \textbackslash{}chcpd\{\LWR@HTMLsanitize{#2}\}%
29     }{%
30         \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
31     }{% original
32     \group_begin:
33         \tl_if_blank:nF {#2}
34         {
35             \keys_set:nn {chemformula} {#1}
36             \__chemformula_save_catcodes:
37             \__chemformula_sanitize:Nn
38                 \l__chemformula_chemformula_tmpa_tl
39                 {#2}
40             \__chemformula_input_compound_no_check:N
41                 \l__chemformula_compound_tl
42                 \l__chemformula_chemformula_tmpa_tl
43             \__chemformula_prepare_output:N \l__chemformula_compound_tl
44             \chemformula_write:V \l__chemformula_compound_tl
45         }
46     \group_end:
47     }
48     \endgroup
49 }

```

`\charrow` If standalone, appears in a regular lateximage.

```

50 \RenewDocumentCommand \charrow { m0{}0{} }
51 {
52   \begin{lateximage}[(-chemformula- charrow)]
53   \group_begin:
54   \__chemformula_draw_arrow:nnn {#1} {#2} {#3}
55   \group_end:
56   \end{lateximage}
57 }

```

\chname If standalone, appears in a regular lateximage, hashed according to contents.

```

58 \RenewDocumentCommand \chname { R(){}R(){} }
59 {
60   \begin{lateximage}*[%
61     \textbackslash{}chname(\LWR@HTMLsanitize{#1})(\LWR@HTMLsanitize{#2})
62   ]%
63   \chemformula_chwritebelow:nn {#1} {#2}
64   \end{lateximage}
65 }

```

\chlewis Placed inline, hashed according to contents and options.

```

66 \RenewDocumentCommand \chlewis { 0{}mm }
67 {
68   \begingroup%
69   \boolfalse{mathjax}%
70   \LWR@subsingledollar*{\textbackslash{}chlewis\{#2\}\{#3\}}%
71   {
72     \protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
73   }{
74     \chemformula_lewis:nnn {#1} {#2} {#3}
75   }
76   \endgroup%
77 }

```

lwarp redefines the \$ character, so special handling is required to escape math expressions inside \ch.

This boolean tracks a new kind of escaped math:

```

78 \bool_new:N      \l__chemformula_first_last_LWRdollar_bool

```

\chemformula_input_escape_math

Adds additional escaping for the new dollar definition:

```

79 \cs_gset_protected:Npn \__chemformula_input_escape_math:n #1
80 {

```

```

81  \_chemformula_first_last_math:n {#1}
82  \bool_if:NT \l__chemformula_first_last_dollar_bool
83    {
84      \bool_set_true:N \l__chemformula_first_last_math_bool
85      \_chemformula_read_escape_dollar:w #1 \q_nil
86    }
87  \bool_if:NT \l__chemformula_first_last_mathbraces_bool
88    {
89      \bool_set_true:N \l__chemformula_first_last_math_bool
90      \_chemformula_read_escape_mathbraces:w #1 \q_nil
91    }

```

Added by **lwarp**:

```

92  \bool_if:NT \l__chemformula_first_last_LWRdollar_bool%      lwarp
93    {
94      \bool_set_true:N \l__chemformula_first_last_math_bool%  lwarp
95      \_chemformula_read_escape_LWRdollar:w #1 \q_nil%        lwarp
96    }
97  }

```

`\chemformula_read_escape_LWRdollar`

The following parses the contents inside the new dollars.

lwarp keeps the dollar as its original math shift until the document starts. While **chemmacros** is being patched, the dollar must temporarily be set to its new meaning during the following definition.

```

98 \begingroup
99 \catcode'\$=\active
100
101 \cs_new_protected:Npn \_chemformula_read_escape_LWRdollar:w $#1$ \q_nil
102 {
103   \_chemformula_read_escape_math:n {#1}
104 }
105
106 \endgroup

```

`\chemformula_bool_set_if_first_last`

The following looks at the first and last tokens for delimiters to escape math inside `\ch`. The original definition is modified to look for the control sequences which are used by the new meaning of `$`.

```

107 \cs_new_protected:Npn \_chemformula_bool_cs_set_if_first_last:NnNN #1#2#3#4
108 {
109   \int_zero:N \l__chemformula_tmpa_int
110   \int_zero:N \l__chemformula_tmpb_int

```

```

111 \int_set:Nn \l__chemformula_tmpa_int { \tl_count:n {#2} }
112 \tl_map_inline:nn {#2}
113 {
114   \int_incr:N \l__chemformula_tmpb_int
115   \int_compare:nT { \l__chemformula_tmpb_int = 1 }
116   {

```

At the start, the cs_ version compares control sequences:

```

117   \ifdefstrequal{##1}{#3}% lwarp
118   {
119     \bool_set_true:N #1
120   }% lwarp
121   {}
122 }

```

At the end, compare more control sequences:

```

123   \int_compare:nT { \l__chemformula_tmpb_int = \l__chemformula_tmpa_int }
124   {
125     \ifdefstrequal{##1}{#4}
126     {}
127     {
128       \bool_set_false:N #1
129     }
130   }
131 }
132 }

```

\chemformula_first_last_math

Modified to check for the new meaning of \$ at first/last:

```

133 \cs_gset_protected:Npn \__chemformula_first_last_math:n #1
134 {
135   \bool_set_false:N \l__chemformula_first_last_math_bool
136   \bool_set_false:N \l__chemformula_first_last_dollar_bool
137   \bool_set_false:N \l__chemformula_first_last_LWRdollar_bool% lwarp
138   \bool_set_false:N \l__chemformula_first_last_mathbraces_bool
139   \__chemformula_bool_set_if_first_last:Nnnn
140   \l__chemformula_first_last_dollar_bool
141   {#1}
142   { $ } { $ }
143   \bool_if:NF \l__chemformula_first_last_dollar_bool
144   {
145     \__chemformula_bool_set_if_first_last:Nnnn
146     \l__chemformula_first_last_mathbraces_bool
147     {#1}
148     { \ ( ) { \ ) }

```

Added by lwarp:

```

149      \bool_if:NF \l__chemformula_first_last_mathbraces_bool%   lwarp
150      {
151          \__chemformula_bool_cs_set_if_first_last:NnNN
152          \l__chemformula_first_last_LWRdollar_bool
153          {#1}
154          { \LWR@newsingledollar } { \LWR@newsingledollar }
155      }%   lwarp
156  }
157 }

158 \ExplSyntaxOff

```

File 49 `lwarp-chemgreek.sty`

§ 141 Package **chemgreek**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg `chemgreek` **chemgreek** is patched for use by **lwarp**.

Greek symbols

⚠ package selection

To use text-mode symbols, use packages **textalpha** or **textgreek**. Using the other packages supported by **chemgreek** will result in math-mode greek characters, which will result in svg images being used. These images will be hashed.

⚠ Xe_{La}TeX, Lua_{La}TeX

If using Xe_{La}TeX or Lua_{La}TeX, select the fontspec mapping:

```
\selectchemgreekmapping{fontspec}
```

for HTML output:

```

1 \LWR@ProvidesPackagePass{chemgreek}[2016/02/10]

2 \ExplSyntaxOn
3
4 \cs_gset_protected:Npn \chemgreek_text:n #1
5   { { \text {#1} } }
6
7 \appto\LWR@restoreorigformatting{%
8 \cs_set_protected:Npn \chemgreek_text:n #1%
9   { \ensuremath { \text {#1} } } }%
10 }
11
12 \ExplSyntaxOff

```

File 50 **lwarp-chemmacros.sty**

§ 142 Package **chemmacros**


(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg chemmacros **chemmacros** is patched for use by **lwarp**.


for HTML output: `1 \LWR@ProvidesPackagePass{chemmacros}`

SVG file hashing assumes that the relevant options are constant for the entire document.

§ 142.1 **Changes to the user's document**

 **\makepolymerdelims** When using `\makepolymerdelims`, enclose the entire expression inside a `polymerdelims` environment, such as (from the **chemmacros** manual):

```
\begin{polymerdelims}
\chemfig{-[@{op,.75}]CH_2-CH(-[6]Cl)-[@{c1,0.25}]}
\makepolymerdelims{5pt}[27pt]{op}{c1}
\end{polymerdelims}
```

 **redox reactions** Redox reactions must be enclosed inside a `redoxreaction` environment. For print output, extra space must be included above and/or below the result, so they are declared as arguments to the environment, instead of being manually entered as per the **chemmacros** manual. For HTML output, the extra space is ignored and a `lateximage` is used instead.

```
\begin{redoxreaction}{7mm}{7mm}
\OX{a,Na} $\rightarrow$ \OX{b,Na}\pch{redox(a,b){oxidation}}
\end{redoxreaction}
```

§ 142.2 **Code**§ 142.3 **Loading modules**

Patching **chemmacros** modules must be done `\AtBeginDocument`, since modules are invoked by the user in the preamble, and each patch is only done if the module is loaded.

```

2 \ExplSyntaxOn
3
4 \newcommand{\@ifchemmacrosmoduleloaded}[1]{%
5 \@ifl@aded{\c__chemmacros_module_extension_tl}{\c__chemmacros_module_prefix_tl.#1}%
6 }
7
8 \ExplSyntaxOff

```

§ 142.4 **New environments**

`\makepolymerdelims` and redox reactions must be enclosed in a `lateximage` during HTML output. These environments are provided here in HTML mode, and in the **lwarp** core in print mode, as a high-level semantic syntax which automatically embeds the contents in a `lateximage` with an appropriate alt tag.

Env **polymerdelims**

```

9 \DeclareDocumentEnvironment{polymerdelims}{}
10 {\begin{lateximage}[(-chemmacros- polymer)]}
11 {\end{lateximage}}

```

Env **redoxreaction** $\{\langle space\ above\rangle\} \{\langle space\ below\rangle\}$

For HTML output, the above and below space is ignored, and a `lateximage` is used instead. For the print output version, see section [82](#).

```

12 \DeclareDocumentEnvironment{redoxreaction}{m m}
13 {\begin{lateximage}[(-chemmacros- redoxreaction)]}
14 {\end{lateximage}}

15 \ExplSyntaxOn

```

§ 142.5 Acid-base

```

16 \AtBeginDocument{
17 \@ifchemmacrosmoduleloaded{acid-base}{
18 \PackageInfo{lwarp}{Patching~chemmacros~module~acid-base}
19
20 \cs_gset_protected:Npn \chemmacros_p:n #1
21   {
22     \begingroup
23     \boolfalse{mathjax}
24     \LWR@subsingledollar*{
25       \textbackslash{}p{\LWR@HTMLsanitize{#1}\}
26     }{
27       chemmacrosp\protect\LWR@HTMLsanitize{\detokenize\expandafter{#1}}%
28     }{
29       \group_begin:
30       \mbox
31       {
32         \chemmacros_p_style:n {p}
33         \ensuremath {#1}
34       }
35       \group_end:
36     }
37     \endgroup
38   }
39
40 \RenewDocumentCommand \pH {} {
41   \begingroup
42   \boolfalse{mathjax}
43   \LWR@subsingledollar*{\textbackslash{}pH}{chemmacros}{
44     \chemmacros_p:n { \chemmacros_chemformula:n {H} }
45   }
46   \endgroup
47 }
48
49 \RenewDocumentCommand \pOH {} {
50   \begingroup
51   \boolfalse{mathjax}
52   \LWR@subsingledollar*{\textbackslash{}pOH}{chemmacros}{
53     \chemmacros_p:n { \chemmacros_chemformula:n {OH} }
54   }
55   \endgroup
56 }
57
58 \RenewDocumentCommand \pKa {0{}}
59   {
60     \begingroup
61     \boolfalse{mathjax}
62     \LWR@subsingledollar*{\textbackslash{}pKa{[]#1{[]}}}{chemmacros #1}{
63       \chemmacros_p:n

```

```

64      {
65          \Ka \ifblank {#1} {}
66          { {} \c_math_subscript_token { \chemmacros_bold:n {#1} } }
67      }
68  }
69  \endgroup
70 }
71
72 \RenewDocumentCommand \pKb {0{}}
73 {
74     \begingroup
75     \boolfalse{mathjax}
76     \LWR@subsingledollar*{\textbackslash{}pKb{[]#1{[]}}}{chemmacros #1}{
77         \chemmacros_p:n
78         {
79             \Kb \ifblank {#1} {}
80             { {} \c_math_subscript_token { \chemmacros_bold:n {#1} } }
81         }
82     }
83     \endgroup
84 }
85
86 \LetLtxMacro\LWR@chemmacros@origKa\Ka
87 \renewcommand*{\Ka}{%
88     \begingroup
89     \boolfalse{mathjax}
90     \LWR@subsingledollar*{\textbackslash{}Ka}{chemmacros}{%
91         \LWR@chemmacros@origKa%
92     }%
93     \endgroup
94 }
95
96 \LetLtxMacro\LWR@chemmacros@origKb\Kb
97 \renewcommand*{\Kb}{%
98     \begingroup
99     \boolfalse{mathjax}
100     \LWR@subsingledollar*{\textbackslash{}Kb}{chemmacros}{%
101         \LWR@chemmacros@origKb%
102     }%
103     \endgroup
104 }
105
106 \LetLtxMacro\LWR@chemmacros@origKw\Kw
107 \renewcommand*{\Kw}{%
108     \begingroup
109     \boolfalse{mathjax}
110     \LWR@subsingledollar*{\textbackslash{}Kw}{chemmacros}{
111         \LWR@chemmacros@origKw
112     }
113     \endgroup

```

```

114 }
115
116 }{}% \@ifchemmacrosmoduleloaded
117 }% AtBeginDocument

```

§ 142.6 Charges

```

118 \AtBeginDocument{
119 \@ifchemmacrosmoduleloaded{charges}{
120 \PackageInfo{lwarp}{Patching~chemmacros~module~charges}
121
122 \cs_gset_protected:Npn \fplus {
123   \begingroup
124   \boolfalse{mathjax}
125   \LWR@subsingledollar*{\textbackslash}{fplus}{chemmacros}
126   { \LWR@origensuredmath{\chemformula_fplus:} }
127   \endgroup
128 }
129 \cs_gset_protected:Npn \fminus {
130   \begingroup
131   \boolfalse{mathjax}
132   \LWR@subsingledollar*{\textbackslash}{fminus}{chemmacros}
133   { \LWR@origensuredmath{\chemformula_fminus:} }
134   \endgroup
135 }
136
137 }{}% \@ifchemmacrosmoduleloaded
138 }% AtBeginDocument

```

§ 142.7 Nomenclature

```

139 \AtBeginDocument{
140 \@ifchemmacrosmoduleloaded{nomenclature}{
141 \PackageInfo{lwarp}{Patching~chemmacros~module~nomenclature}
142
143 \cs_gset_protected:Npn \chemmacros_charge:n #1
144 {
145   \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}{
146     {\chemmacros_chemformula:n { {}^{#1} }}
147     {
148       \ifmmode
149         {\chemmacros_chemformula:n { {}^{#1} }}
150       \else
151         { \textsuperscript{\ensuremath{#1}} }
152       \fi
153     }
154   }
155
156

```

```
157 \LetLtxMacro\LWR@chemmacros@origchemprime\chemprime
158
159 \protected\def\chemprime { \HTMLUnicode{2032} }
160
161 \appto\LWR@restoreorigformatting{%
162 \LetLtxMacro\chemprime\LWR@chemmacros@origchemprime%
163 }

164 \ChemCompatibilityFrom{5.8}
165 \cs_gset_protected:Npn \__chemmacros_cip:n #1
166 {
167   \tl_set:Nn \l__chemmacros_tmpa_tl {#1}
168   \int_step_inline:nnnn {0} {1} {9}
169   {
170     \tl_replace_all:Nnn \l__chemmacros_tmpa_tl
171     {##1}
172     { { \l__chemmacros_cip_number_tl ##1} }
173   }
174   {
175     \l__chemmacros_cip_inner_tl
176     \LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
177       \l__chemmacros_tmpa_tl
178     }}% lwarp
179   }
180 }
181 \EndChemCompatibility

182 \RenewDocumentCommand \Sconf { 0{S} } {
183 \begin{lateximage}[\textbackslash{}Sconf{[]#1{}}]
184   \chemmacros_sconf:n {#1}
185 \end{lateximage}
186 }
187
188 \RenewDocumentCommand \Rconf { 0{R} } {
189 \begin{lateximage}[\textbackslash{}Rconf{[]#1{}}]
190   \chemmacros_rconf:n {#1}
191 \end{lateximage}
192 }

193 \cs_gset_protected:Npn \chemmacros_hapto:n #1
194 {
195   \begingroup
196   \boolfalse{mathjax}
197   \LWR@subsingledollar*{\textbackslash{}hapto\{#1\}}{\chemmacros}{
198     \chemmacros_coordination_symbol:nnnn
199     { \l__chemmacros_coord_use_hyphen_bool }
200     {
201       \chemmacros_if_compatibility:nnTF {>} {5.7}
202       { \c_true_bool }
203       { \c_false_bool }
```

```

204     }
205     { \chemeta }
206     {#1}
207   }
208   \endgroup
209 }
210
211 \cs_gset_protected:Npn \chemmacros_dento:n #1
212 {
213   \begingroup
214   \boolfalse{mathjax}
215   \LWR@subsingledollar*{\textbackslash{}dento\{#1\}}{\chemmacros}{
216     \chemmacros_coordination_symbol:nnnn
217     { \l__chemmacros_coord_use_hyphen_bool }
218     {
219       \chemmacros_if_compatibility:nnTF {>} {5.7}
220       { \c_true_bool }
221       { \c_false_bool }
222     }
223     { \chemkappa }
224     {#1}
225   }
226   \endgroup
227 }
228
229 \cs_gset_protected:Npn \chemmacros_bridge:n #1
230 {
231   \begingroup
232   \boolfalse{mathjax}
233   \LWR@subsingledollar*{\textbackslash{}bridge\{#1\}}{\chemmacros}{
234     \chemmacros_coordination_symbol:nnnn
235     { \l__chemmacros_coord_use_hyphen_bool }
236     { \l__chemmacros_bridge_super_bool }
237     { \chemmu }
238     {#1}
239   }
240   \endgroup
241 }
242 }{}% \@ifchemmacrosmoduleloaded
243 }% AtBeginDocument

```

§ 142.8 Particles

```

244 \AtBeginDocument{
245   \@ifchemmacrosmoduleloaded{particles}{
246     \PackageInfo{lwarp}{Patching~chemmacros~module~particles}
247   }
248   \cs_gset_protected:Npn \chemmacros_declare_nucleophile:Nn #1#2
249   {

```

```

250 \cs_set_protected:cpn {__chemmacros_ \chemmacros_remove_backslash:N #1:}
251 {
252   \bool_if:NTF \l__chemmacros_nucleophile_elpair_bool
253   {
254     \chemmacros_elpair:n { #2 }
255     \chemmacros_if_compatibility:nnT {>=} {5.3}
256     { \skip_horizontal:N \l__chemmacros_nucleophile_dim }
257     \chemmacros_chemformula:n { {}^{}- }
258   }
259   { \chemmacros_chemformula:n { #2^{}- } }
260 }
261 \DeclareDocumentCommand #1 {o}
262 {%
263   \begin{lateximage}%
264   \group_begin:%
265   \IfNoValueF {##1}%
266   { \chemmacros_set_keys:nn {particles} {##1} }%
267   \use:c {__chemmacros_ \chemmacros_remove_backslash:N #1:}%
268   \group_end:%
269   \end{lateximage}%
270 }
271 }
272
273 \RenewChemNucleophile \Nuc {Nu}
274 \RenewChemNucleophile \ba {ba}
275
276 }{}% \@ifchemmacrosmoduleloaded
277 }% AtBeginDocument

```

§ 142.9 Phases

```

278 \AtBeginDocument{
279 \@ifchemmacrosmoduleloaded{phases}{
280 \PackageInfo{lwarp}{Patching~chemmacros~module~phases}
281
282 \cs_undefine:N \chemmacros_phase:n
283 \cs_new_protected:Npn \chemmacros_phase:n #1
284 {
285   \chemmacros_leave_vmode:
286   \bool_if:NTF \l__chemmacros_phases_sub_bool
287   {
288     \ifnumequal{\value{LWR@lateximagedepth}}{0}
289     {
290       \textsubscript{ (#1) }
291     }
292     {
293       \chemformula_subscript:n { (#1) }
294     }
295   }

```

```

296     {
297       \skip_horizontal:N \l__chemmacros_phases_space_dim
298       \chemmacros_text:n { (#1) }
299     }
300   }
301
302 }{}% \@ifchemmacrosmoduleloaded
303 }% AtBeginDocument

```

§ 142.10 Mechanisms

```

304 \AtBeginDocument{
305 \@ifchemmacrosmoduleloaded{mechanisms}{
306 \PackageInfo{lwarp}{Patching~chemmacros~module~mechanisms}
307
308 \chemmacros_define_keys:nn {textmechanisms}
309   {
310     type      .choice: ,
311     type /    .code:n   =
312     {
313       \__chemmacros_set_mechanisms:nnn { S }
314       {
315         \textsubscript{N}
316       }
317       { }
318     } ,
319     type / 1 .code:n   =
320     {
321       \__chemmacros_set_mechanisms:nnn { S }
322       {
323         \textsubscript{N}
324         1
325       }
326       { }
327     } ,
328     type / 2 .code:n   =
329     {
330       \__chemmacros_set_mechanisms:nnn { S }
331       {
332         \textsubscript{N}
333         2
334       }
335       { }
336     } ,
337     type / se .code:n  =
338     {
339       \__chemmacros_set_mechanisms:nnn { S }
340       {
341         \textsubscript{E}

```

```

342     }
343     { }
344   } ,
345   type / 1e .code:n    =
346   {
347     \__chemmacros_set_mechanisms:nnn { S }
348     {
349       \textsubscript{E}
350       1
351     }
352     { }
353   } ,
354   type / 2e .code:n    =
355   {
356     \__chemmacros_set_mechanisms:nnn { S }
357     {
358       \textsubscript{E}
359       2
360     }
361     { }
362   } ,
363   type / ar .code:n    =
364   {
365     \__chemmacros_set_mechanisms:nnn { S }
366     {
367       \textsubscript{E}
368     }
369     { Ar - }
370   } ,
371   type / e .code:n     =
372   { \__chemmacros_set_mechanisms:nnn { E } { } { } } ,
373   type / e1 .code:n    =
374   { \__chemmacros_set_mechanisms:nnn { E } { 1 } { } } ,
375   type / e2 .code:n    =
376   { \__chemmacros_set_mechanisms:nnn { E } { 2 } { } } ,
377   type / cb .code:n    =
378   {
379     \__chemmacros_set_mechanisms:nnn { E }
380     {
381       1
382       \textsubscript{cb}
383     }
384     { }
385   } ,
386   type .default:n =
387 }
388
389 \cs_gset_protected:Npn \chemmacros_mechanisms:n #1
390 {
391   \tl_if_blank:nTF {#1}

```

```

392     { \chemmacros_set_keys:nn {textmechanisms} { type } }
393     { \chemmacros_set_keys:nn {textmechanisms} { type = #1 } }
394   \mbox
395   {
396     \tl_use:N \l__chemmacros_mechanisms_ar_tl
397     \tl_use:N \l__chemmacros_mechanisms_type_tl
398     \tl_use:N \l__chemmacros_mechanisms_mol_tl
399   }
400 }
401
402 \appto\LWR@restoreorigformatting{%
403 \cs_set_protected:Npn \chemmacros_mechanisms:n #1%
404   {%
405     \tl_if_blank:nTF {#1}%
406     { \chemmacros_set_keys:nn {mechanisms} { type } }%
407     { \chemmacros_set_keys:nn {mechanisms} { type = #1 } }%
408   \mbox%
409   {%
410     \tl_use:N \l__chemmacros_mechanisms_ar_tl%
411     \tl_use:N \l__chemmacros_mechanisms_type_tl%
412     \tl_use:N \l__chemmacros_mechanisms_mol_tl%
413   }%
414 }%
415 }
416
417 }{}% \@ifchemmacrosmoduleloaded
418 }% AtBeginDocument

```

§ 142.11 Newman

```

419 \AtBeginDocument{
420 \@ifchemmacrosmoduleloaded{newman}{
421 \PackageInfo{lwarp}{Patching~chemmacros~module~newman}
422
423 \RenewDocumentCommand \newman {od()m}%
424   {
425     \IfValueTF{#2}
426     {\begin{lateximage}[\textbackslash{}newman{#2}\{#3\}]}
427     {\begin{lateximage}[\textbackslash{}newman\{#3\}]}
428     \group_begin:
429     \IfNoValueF {#1} { \chemmacros_set_keys:nn {newman} {#1} }
430     \IfNoValueTF {#2}
431     { \chemmacros_newman:nn { } {#3} }
432     { \chemmacros_newman:nn {#2} {#3} }
433     \group_end:
434     \end{lateximage}
435   }%
436
437 }{}% \@ifchemmacrosmoduleloaded

```

```
438}% AtBeginDocument
```

§ 142.12 Orbital

```
439 \AtBeginDocument{
440 \ifchemmacrosmoduleloaded{orbital}{
441 \PackageInfo{lwarp}{Patching~chemmacros~module~orbital}
442
443 \RenewDocumentCommand \orbital {om}
444 {
445   \IfValueTF{#1}
446   {
447     \begin{lateximage}[%
448       \textbackslash{}orbital{[]\LWR@HTMLsanitize{#1}-{}}\{#2\}%
449     ][] [margin-left: 1em ; margin-right: 1em]
450   }
451   {
452     \begin{lateximage}[%
453       \textbackslash{}orbital\{#2\}%
454     ][] [margin-left: 1em ; margin-right: 1em]
455   }
456   \group_begin:
457   \chemmacros_set_keys:nn {orbital/type} {#2}
458   \IfNoValueTF {#1}
459   { \chemmacros_orbital:n { } }
460   { \chemmacros_orbital:n {#1} }
461   \group_end:
462   \end{lateximage}
463 }
464
465 }{}% \ifchemmacrosmoduleloaded
466}% AtBeginDocument
```

§ 142.13 Reactions

`\chemmacros_declare_reaction_env` {<chem>} {<math>} {<args number>} {<argument list ({#2}{#3}...)}{}

```
467 \AtBeginDocument{
468 \ifchemmacrosmoduleloaded{reactions}{
469 \PackageInfo{lwarp}{Patching~chemmacros~module~orbital}
470
471 \cs_gset_protected:Npn \chemmacros_declare_reaction_env:nnnn #1#2#3#4
472 {
473   \exp_args:Nnx \DeclareDocumentEnvironment {#1} { 0 } \prg_replicate:nn {#3+0} {m} }
474 {
475   \boolfalse{mathjax}% lwarp
476   \chemmacros_add_reaction_description:n {##1}
477   \__chemmacros_begin_reaction:
478   \chemmacros_reaction_read:nnw {#2} {#4}
```

```

479     }
480     {
481         \_chemmacros_end_reaction:
482     }
483 }
484 \cs_generate_variant:Nn \chemmacros_declare_reaction_env:nnnn {nnnV}
485
486 \RenewChemReaction {reaction} {equation}
487 \RenewChemReaction {reaction*} {equation*}
488 \RenewChemReaction {reactions} {align}
489 \RenewChemReaction {reactions*} {align*}
490
491 }{}% \@ifchemmacrosmoduleloaded
492 }% AtBeginDocument

```

§ 142.14 Redox

```

493 \AtBeginDocument{
494 \@ifchemmacrosmoduleloaded{redox}{
495 \PackageInfo{lwarp}{Patching~chemmacros~module~redox}
496
497 \NewDocumentCommand \LWR@chemmacros@ox { s m >{\SplitArgument{1}{,}}m }
498 {
499     \IfBooleanTF {#1}
500     { \chemmacros_ox:nnnn {#1} {#2} #3 }
501     { \chemmacros_ox:nnnn { } {#2} #3 }
502 }
503
504 \RenewDocumentCommand \ox { s O{ } m }
505 {
506     \begingroup
507     \boolfalse{mathjax}
508     \IfBooleanTF {#1}
509     {
510         \LWR@subsingledollar*{% yes hash
511             \textbackslash{}ox*\{\LWR@HTMLsanitize{#3}\}% alt
512         }{%
513             star \protect\LWR@HTMLsanitize{\detokenize\expandafter{#2}}}%
514         }{%
515             \LWR@chemmacros@ox* {#2} {#3}% contents
516         }%
517     }
518     {
519         \LWR@subsingledollar*{% yes hash
520             \textbackslash{}ox*\{\LWR@HTMLsanitize{#3}\}% alt
521         }{%
522             \protect\LWR@HTMLsanitize{\detokenize\expandafter{#2}}}%
523         }{%
524             \LWR@chemmacros@ox {#2} {#3}% contents
525         }%

```

```

526     }
527   \endgroup
528 }
529
530 }{}% \@ifchemmacrosmoduleloaded
531 }% AtBeginDocument

```

§ 142.15 Scheme

Fix for **chemmacros** as of v5.8b, when using **newfloat** and **babel**:

```

532 \AtBeginDocument{
533 \@ifchemmacrosmoduleloaded{scheme}{
534 \PackageInfo{lwarp}{Patching~chemmacros~module~scheme}
535
536 \ifdefstring{\schemename}{los}{
537 \SetupFloatingEnvironment{scheme}{
538 name = \chemmacros_translate:n {scheme-name}
539 }
540 }{}
541
542 }{}% \@ifchemmacrosmoduleloaded
543 }% AtBeginDocument

```

§ 142.16 Spectroscopy

```

544 \AtBeginDocument{
545 \@ifchemmacrosmoduleloaded{spectroscopy}{
546 \PackageInfo{lwarp}{Patching~chemmacros~module~spectroscopy}
547
548 \ChemCompatibilityTo{5.8}
549 \cs_gset_protected:Npn \__chemmacros_nmr_base:nn #1#2
550 {
551   \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
552   {
553     \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { \{ }
554     \tl_put_right:Nn \g__chemmacros_nmr_element_coupled_tl { \} }
555   }
556   \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl {#2}
557   % \chemmacros_chemformula:n { ^{#1} }
558   \textsuperscript{#1}
559   \bool_if:NTF \l__chemmacros_nmr_parse_bool
560   { \chemformula_ch:nV {} \g__chemmacros_nmr_element_coupled_tl }
561   { \chemmacros_chemformula:V \g__chemmacros_nmr_element_coupled_tl }
562   \tl_use:N \l__chemmacros_nmr_element_method_connector_tl
563   \tl_use:N \l__chemmacros_nmr_method_tl
564 }
565 \EndChemCompatibility
566 \ChemCompatibilityFrom{5.8}

```

```
567 \cs_gset_protected:Npn \__chemmacros_nmr_base:nn #1#2
568 {
569   \group_begin:
570   \tl_use:N \l__chemmacros_nmr_base_format_tl
571   \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
572   {
573     \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl { \{ }
574     \tl_put_right:Nn \g__chemmacros_nmr_element_coupled_tl { \} }
575   }
576   \tl_put_left:Nn \g__chemmacros_nmr_element_coupled_tl {#2}
577 %   \chemmacros_chemformula:n { ^{#1} }
578   \textsuperscript{#1}
579   \tl_if_blank:VF \g__chemmacros_nmr_element_coupled_tl
580   {
581     \bool_if:NTF \l__chemmacros_nmr_parse_bool
582     { \chemformula_ch:nV {} \g__chemmacros_nmr_element_coupled_tl }
583     { \chemmacros_chemformula:V \g__chemmacros_nmr_element_coupled_tl }
584   }
585   \tl_use:N \l__chemmacros_nmr_element_method_connector_tl
586   \tl_use:N \l__chemmacros_nmr_method_tl
587   \group_end:
588 }
589 \EndChemCompatibility
590
591
592 \cs_gset_protected:Npn \chemmacros_nmr_position:n #1
593 {
594   \chemmacros_chemformula:x
595   {
596     \exp_not:V \g__chemmacros_nmr_element_tl
597     \bool_if:NF \l__chemmacros_nmr_position_side_bool
598     {
599       \tl_if_eq:NnTF \l__chemmacros_nmr_position_tl {^}% lwarp
600       { \textsuperscript{\exp_not:n { {#1} }} }% lwarp
601       { \textsubscript{\exp_not:n { {#1} }} }% lwarp
602 %     \exp_not:V \l__chemmacros_nmr_position_tl
603 %     \exp_not:n { {#1} }
604     }
605   }
606   \bool_if:NT \l__chemmacros_nmr_position_side_bool
607   {
608     \tl_use:N \l__chemmacros_nmr_position_tl
609     \__chemmacros_nmr_position:n {#1}
610   }
611 }
612
613 \cs_gset_protected:Npn \__chemmacros_nmr_coupling:w (#1;#2)
614 {
615   \tl_set:Nn \l__chemmacros_nmr_coupling_bonds_tl
616   {
```

```

617     \l__chemmacros_nmr_coupling_bonds_pre_tl
618     #1
619     \l__chemmacros_nmr_coupling_bonds_post_tl
620   }
621   \bool_if:NTF \l__chemmacros_nmr_coupling_nuclei_sub_bool
622   {
623     \tl_set:Nn \l__chemmacros_nmr_coupling_nuclei_tl
624     {
625       \c_math_subscript_token
626       \textsubscript% lwarp
627       {
628         \l__chemmacros_nmr_coupling_nuclei_pre_tl
629         \chemmacros_chemformula:n {#2}
630         \l__chemmacros_nmr_coupling_nuclei_post_tl
631       }
632     }
633   }
634   {
635     \tl_set:Nn \l__chemmacros_nmr_coupling_nuclei_tl
636     {
637       \l__chemmacros_nmr_coupling_nuclei_pre_tl
638       \chemmacros_chemformula:n {#2}
639       \l__chemmacros_nmr_coupling_nuclei_post_tl
640     }
641   }
642   \__chemmacros_nmr_coupling_aux_i:w
643 }
644
645 \AfterEndPreamble{% After \AtBeginDocument
646 % \NMR{<num>,<elem>}{<num>,<unit>}[<solvent>] ALL arguments are optional
647 % \NMR* same but without ": $\delta$" at end
648 \cs_gset_protected:Npn \chemmacros_nmr:nnnn #1#2#3#4
649 {
650   \bool_if:NT \l__chemmacros_nmr_list_bool { \item \scan_stop: }
651   \group_begin:
652     \chemmacros_leave_vmode:
653     \bool_set_false:N \l__chemmacros_nmr_frequency_bool
654     \bool_set_false:N \l__chemmacros_nmr_solvent_bool
655     \tl_if_empty:nF {#3}
656     { \bool_set_true:N \l__chemmacros_nmr_frequency_bool }
657     \tl_if_empty:nF {#4}
658     { \bool_set_true:N \l__chemmacros_nmr_solvent_bool }
659     \bool_if:nT
660     {
661       \l__chemmacros_nmr_frequency_bool
662       ||
663       \l__chemmacros_nmr_solvent_bool
664     }
665     { \bool_set_true:N \l__chemmacros_nmr_delimiters_bool }
666     \bool_if:nT

```

```

667     {
668         \l__chemmacros_nmr_frequency_bool
669         &&
670         \l__chemmacros_nmr_solvent_bool
671     }
672     { \bool_set_true:N \l__chemmacros_nmr_comma_bool }
673     \tl_if_empty:nTF {#2}
674     {
675         \__chemmacros_nmr_nucleus:VV
676         \l__chemmacros_nmr_isotope_default_tl
677         \l__chemmacros_nmr_element_default_tl
678     }
679     { \__chemmacros_nmr_nucleus:w #2 \q_stop }
680     \mode_if_math:TF
681     {
682         \text
683         {
684             \group_begin:
685             \tl_use:N \l__chemmacros_nmr_format_tl
686             \LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
687                 \__chemmacros_nmr_base:VV
688                 \g__chemmacros_nmr_isotope_tl
689                 \g__chemmacros_nmr_element_tl
690                 \bool_if:NT \l__chemmacros_nmr_delimiters_bool
691                 { ~ ( }
692                 \bool_if:NT \l__chemmacros_nmr_frequency_bool
693                 { \__chemmacros_nmr_frequency:n {#3} }
694                 \bool_if:NT \l__chemmacros_nmr_comma_bool
695                 { , ~ }
696                 \bool_if:NT \l__chemmacros_nmr_solvent_bool
697                 { \chemmacros_chemformula:n {#4} }
698                 \bool_if:NT \l__chemmacros_nmr_delimiters_bool
699                 { ) }
700                 \tl_if_blank:nT {#1} {::~}
701             }}% lwarp
702             \group_end:
703         }
704         \tl_if_blank:nT {#1}
705         {
706             \delta
707             \text { \l__chemmacros_nmr_delta_tl }
708             \bool_if:NT \l__chemmacros_nmr_use_equal_bool {=}
709         }
710     }
711     {
712         \group_begin:
713         \tl_use:N \l__chemmacros_nmr_format_tl
714         \LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
715             \__chemmacros_nmr_base:VV
716             \g__chemmacros_nmr_isotope_tl

```

```

717         \g__chemmacros_nmr_element_tl
718     \bool_if:NT \l__chemmacros_nmr_delimiters_bool
719         {~{}}
720     \bool_if:NT \l__chemmacros_nmr_frequency_bool
721         { \__chemmacros_nmr_frequency:n {#3} }
722     \bool_if:NT \l__chemmacros_nmr_comma_bool
723         {,~}
724     \bool_if:NT \l__chemmacros_nmr_solvent_bool
725         {
726             \bool_if:NTF \l__chemmacros_nmr_parse_bool
727                 { \chemformula_ch:nn { } {#4} }
728                 {#4}
729         }
730     \bool_if:NT \l__chemmacros_nmr_delimiters_bool
731         {}
732 }}% lwarp
733 \tl_if_blank:nT {#1} {:}
734 \group_end:
735 \tl_if_blank:nT {#1}
736 {
737     \tl_use:N \c_space_tl
738     \c_math_toggle_token
739     \delta
740     \c_math_toggle_token
741     \l__chemmacros_nmr_delta_tl
742     \bool_if:NT \l__chemmacros_nmr_use_equal_bool {~=}
743 }
744 }
745 \group_end:
746 }
747 }% AfterEndPreamble
748
749
750 \RenewDocumentCommand \chemmacros_data:w { smo }
751 {
752     \bool_if:NT \l__chemmacros_nmr_list_bool { \item }
753     {
754 %         \tl_use:N \l__chemmacros_nmr_format_tl #2
755         \tl_use:N \l__chemmacros_nmr_format_tl
756         \LWR@textcurrentcolor{\LWR@textcurrentfont{% lwarp
757             #2
758             \IfNoValueF {#3} { ~ ( #3 ) }
759             \IfBooleanT {#1} { \bool_if:NT \l__chemmacros_nmr_use_equal_bool { : } }
760         }}% lwarp
761     }
762     \IfBooleanF {#1} { \bool_if:NT \l__chemmacros_nmr_use_equal_bool { ~ = } }
763 }
764
765 }{}% \@ifchemmacrosmoduleloaded
766 }% AtBeginDocument

```

§ 142.17 Thermodynamics

```

767 \AtBeginDocument{
768 \@ifchemmacrosmoduleloaded{thermodynamics}{
769 \PackageInfo{lwarp}{Patching~chemmacros~module~thermodynamics}
770
771 \cs_gset_protected:Npn \chemmacros_state:nn #1#2
772 {
773   \group_begin:
774   \boolfalse{mathjax}
775   \chemmacros_set_keys:nn {thermodynamics} {#1}
776   \LWR@subsingledollar*{% yes hashing
777     \textbackslash{}state\{\LWR@HTMLSanitize{#2}\}% alt
778   }{%
779     chemmacros_state% add'l hashing
780     #1% options
781     LSP \tl_use:N \l__chemmacros_state_sp_left_tl% super/subscripts
782     LSB \tl_use:N \l__chemmacros_state_sb_left_tl
783     RSP \tl_use:N \l__chemmacros_state_sp_right_tl
784     RSB \tl_use:N \l__chemmacros_state_sb_right_tl
785   }
786   {
787     \LWR@origensuredmath{
788       \chemmacros_text:V \l__chemmacros_state_pre_tl
789       \c_math_superscript_token
790       { \chemmacros_text:V \l__chemmacros_state_sp_left_tl }

```

Only add the subscripts if they are being used. This avoids causing an incorrect depth, as the empty subscript will be measured by \TeX but cropped out by **pdfcrop**.

```

791     \tl_if_empty:NTF \l__chemmacros_state_sb_left_tl
792     {}
793     {
794       \c_math_subscript_token
795       { \chemmacros_text:V \l__chemmacros_state_sb_left_tl }
796     }
797     #2
798     \c_math_superscript_token
799     { \chemmacros_text:V \l__chemmacros_state_sp_right_tl }
800     \tl_if_empty:NTF \l__chemmacros_state_sb_right_tl
801     {}
802     {
803       \c_math_subscript_token
804       { \chemmacros_text:V \l__chemmacros_state_sb_right_tl }
805     }
806     \chemmacros_text:V \l__chemmacros_state_post_tl
807   }
808 }
809 \group_end:
810 }

```

```

811 \cs_generate_variant:Nn \chemmacros_state:nn { nV }
812
813 \cs_gset_protected:Npn \chemmacros_declare_state:Nn #1#2
814 {
815   \chemmacros_define_keys:xn
816     {thermodynamics/\chemmacros_remove_backslash:N #1}
817   {
818     pre .meta:nn = {chemmacros/thermodynamics} { pre = ##1 } ,
819     post .meta:nn = {chemmacros/thermodynamics} { post = ##1 } ,
820     superscript-left .meta:nn = {chemmacros/thermodynamics} { superscript-left = ##1 } ,
821     superscript-right .meta:nn = {chemmacros/thermodynamics} { superscript-right = ##1 } ,
822     superscript .meta:n = { superscript-right = ##1 } ,
823     subscript-left .meta:nn = {chemmacros/thermodynamics} { subscript-left = ##1 } ,
824     subscript-right .meta:nn = {chemmacros/thermodynamics} { subscript-right = ##1 } ,
825     subscript .meta:n = { subscript-left = ##1 } ,
826     subscript-pos .choices:nn =
827       { left , right }
828       { \tl_set_eq:NN \l__chemmacros_state_sb_pos_tl \l_keys_choice_tl } ,
829     symbol .tl_set:N = \l__chemmacros_state_symbol_tl ,
830     unit .tl_set:N = \l__chemmacros_state_unit_tl
831   }
832   \DeclareDocumentCommand #1 { sO{}D(){}m }
833   {
834     \group_begin:
835     \chemmacros_set_keys:xn
836       {thermodynamics/\chemmacros_remove_backslash:N #1}
837       {#2}
838     \tl_if_blank:nF {##3}
839     {
840       \chemmacros_set_keys:nx {thermodynamics}
841       { subscript-\l__chemmacros_state_sb_pos_tl = \exp_not:n {##3} }
842     }
843     \chemmacros_state:nV {##2} \l__chemmacros_state_symbol_tl
844     \chemmacros_set_keys_groups:nnn {thermodynamics} {variables} {##2}
845     \IfBooleanF {##1} { = ~ \SI {##4} { \l__chemmacros_state_unit_tl } }
846   \group_end:
847 }
848 }

```

The pre-existing macros are redefined with the new definition:

```

849 \RenewChemState \enthalpy { symbol = H , unit = \kilo\joule\per\mole }
850 \RenewChemState \entropy { symbol = S , unit = \joule\per\kelvin\per\mole , pre = }
851 \RenewChemState \gibbs { symbol = G , unit = \kilo\joule\per\mole }
852
853 }{}% \@ifchemmacrosmoduleloaded
854 }% AtBeginDocument

855 \ExplSyntaxOff

```

File 51 **lwarp-chemnum.sty**

§ 143 Package **chemnum**

(Emulates or patches code by CLEMENS NIEDERBERGER.)

Pkg chemnum **chemnum** is patched for use by **lwarp**.

for HTML output:

```
1 \LWR@ProvidesPackagePass{chemnum}

2 \ExplSyntaxOn
3
4 \cs_gset_protected:Npn \chemnum_compound_write:n #1
5 {
6   \chemnum_get_compound_property:nn {#1} {pre-main-label-code}
7   \group_begin:
8     \bool_if:NTF \l__chemnum_compound_local_bool
9     { \l__chemnum_local_label_format_tl }
10    { \chemnum_get_compound_property:nn {#1} {label-format} }
11    {
12      \LWR@textcurrentfont{
13        \chemnum_get_compound_property:nn {#1} {counter-representation}
14      }
15    }
16  \group_end:
17  \chemnum_get_compound_property:nn {#1} {post-main-label-code}
18 }
19
20 \cs_gset_protected:Npn \chemnum_subcompound_write:nn #1#2
21 {
22   \group_begin:
23     \bool_if:NTF \l__chemnum_compound_local_bool
24     { \l__chemnum_local_label_format_tl }
25     { \chemnum_get_compound_property:nn {#1} {label-format} }
26     {
27       \LWR@textcurrentfont{
28         \chemnum_get_subcompound_property:nnn {#1} {#2}
29         {counter-representation}
30       }
31     }
32   \group_end:
33 }
34
35 \ExplSyntaxOff
```

File 52 **lwarp-cite.sty**

§ 144 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 53 **lwarp-clrdblpg.sty**

§ 145 Package **clrdblpg**

Pkg clrdblpg **clrdblpg** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{clrdblpg}

File 54 **lwarp-color.sty**

§ 146 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 55 **lwarp-colortbl.sty**

§ 147 Package **colortbl**

Pkg colortbl **colortbl** is emulated.

⚠ **row/cell color** Only use `\rowcolor` and `\cellcolor` at the start of a row, in that order.

colortbl ignores the overhang arguments.

for HTML output: A few placeholder definitions are forgotten first:

```
1 \let\rowcolor\relax
2
3 \LWR@ProvidesPackagePass{colortbl}
```

The following `\LWR@HTML` versions are used inside an HTML tabular.

`\columncolor` [*<model>*] [*<color>*] [*<left overhang>*] [*<right overhang>*]

`\LWR@getmynexttoken` is not used here because `\columncolor` is not used inside the data area of the tabular.

```
4 \NewDocumentCommand{\LWR@HTML@columncolor}{O{named} m o o}{%
5 \convertcolorspec{#1}{#2}{HTML}\LWR@columnHTMLcolor%
6 \LWR@addtabularcellcolor%
7 }
8
9 \LWR@formatted{columncolor}
```

`\LWR@getmynexttoken` is used for `\rowcolor` because it is used inside the data area of the tabular.

`\rowcolor` [*<model>*] [*<color>*] [*<left overhang>*] [*<right overhang>*]

```
10 \NewDocumentCommand{\LWR@HTML@rowcolor}{O{named} m o o}{%
11 \convertcolorspec{#1}{#2}{HTML}\LWR@rowHTMLcolor%
12 \LWR@getmynexttoken%
13 }
14
```

```
15 \LWR@expandableformatted{rowcolor}
```

```
\cellcolor    [⟨model⟩] {⟨color⟩} [⟨left overhang⟩] [⟨right overhang⟩]
```

```
16 \NewDocumentCommand{\LWR@HTML@cellcolor}{0{named} m o o}{%
17 \convertcolorspec{#1}{#2}{HTML}\LWR@cellHTMLcolor%
18 \LWR@addtabularcellcolor%
19 }
20
21 \LWR@formatted{cellcolor}
```

```
\arrayrulecolor  [⟨model⟩] {⟨color⟩}
```

The HTML version for use outside a tabular. Inside a tabular, `\LWR@HTML@arrayrulecolornexttoken` is used instead.

```
22 \newcommand{\LWR@HTML@arrayrulecolor}[2] [named] {%
23 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
24 }
25
26 \LWR@expandableformatted{arrayrulecolor}
```

```
[⟨model⟩] {⟨color⟩}
```

`\LWR@arrayrulecolornexttoken` The HTML version for use inside a tabular.

```
27 \newcommand{\LWR@HTML@arrayrulecolornexttoken}[2] [named] {%
28 \convertcolorspec{#1}{#2}{HTML}\LWR@ruleHTMLcolor%
29 \LWR@getmynexttoken%
30 }
31
32 \LWR@expandableformatted{arrayrulecolornexttoken}
```

```
\doublerulesepcolor  [⟨model⟩] {⟨color⟩}
```

The version for use outside a tabular.

```
33 \newcommand{\LWR@HTML@doublerulesepcolor}[2] [named] {}
34
35 \LWR@expandableformatted{doublerulesepcolor}
```

```
[⟨model⟩] {⟨color⟩}
```

`\LWR@doublerulesepcolornexttoken` The version for use inside a tabular.

```
36 \newcommand{\LWR@HTML@doublerulesepcolornexttoken}[2] [named] {\LWR@getmynexttoken}
37
38 \LWR@expandableformatted{doublerulesepcolornexttoken}
```

File 56 **lwarp-continue.sty**

§ 148 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 57 **lwarp-crop.sty**

§ 149 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 58 **lwarp-cuted.sty**

§ 150 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 59 **lwarp-cutwin.sty**

§ 151 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 60 **lwarp-dblfloatfix.sty**

§ 152 Package **dblfloatfix**

Pkg dblfloatfix **dblfloatfix** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{dblfloatfix}

File 61 **lwarp-dblfnote.sty**

§ 153 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 62 **lwarp-dcolumn.sty**

§ 154 Package **dcolumn**

Pkg dcolumn **dcolumn** is emulated by the **lwarp** core.

1 \LWR@ProvidesPackageDrop{dcolumn}

File 63 **lwarp-diagbox.sty**

§ 155 Package **diagbox**

(Emulates or patches code by LEO LIU.)

Pkg diagbox **diagbox** is patched for use by **lwarp**.

for HTML output: 1 \LWR@ProvidesPackagePass{diagbox}

To restore print-mode inside a lateximage:

```
2 \LetLtxMacro\LWR@origdiagbox@double\diagbox@double
3 \LetLtxMacro\LWR@origdiagbox@triple\diagbox@triple
4
5 \appto\LWR@restoreorigformatting{%
6 \LetLtxMacro\diagbox@double\LWR@origdiagbox@double%
7 \LetLtxMacro\diagbox@triple\LWR@origdiagbox@triple%
8 }
```

```
\LWR@diagbox@AB {\langle E/W \rangle} {\langle A \rangle} {\langle E/W \rangle} {\langle B \rangle}

9 \newcommand{\LWR@diagbox@AB}[4]{
10 \begingroup%
11 \LetLtxMacro\\newline%
12 \BlockClassSingle{diagbox#1}{#2}%
13 \BlockClassSingle{diagbox#3}{#4}%
14 \endgroup%
15 \LWR@stoppars%
16 }
```

```
\LWR@diagboxNW {\langle A \rangle} {\langle B \rangle}

17 \newcommand{\LWR@diagboxNW}[2]{%
18 \LWR@diagbox@AB{E}{#2}{W}{#1}%
19 }
```

Likewise for NE, SW, SE:

```
20 \newcommand{\LWR@diagboxNE}[2]{%
21 \LWR@diagbox@AB{W}{#1}{E}{#2}%
22 }
23
24 \let\LWR@diagboxSW\LWR@diagboxNE
25 \let\LWR@diagboxSE\LWR@diagboxNW
```

```
\diagbox@double  {\langle keys\rangle} {\langle A\rangle} {\langle B\rangle}

26 \def\diagbox@double#1#2#3{%
27 \setkeys{diagbox}{dir=NW,#1}%
28 \csuse{LWR@diagbox\diagbox@dir}{#2}{#3}%
29 }
```

```
\LWR@diagboxTNW  {\langle title\rangle} {\langle A\rangle} {\langle B\rangle}

30 \newcommand{\LWR@diagboxTNW}[3]{%
31 \BlockClassSingle{diagboxtitleN}{#1}
32 \LWR@diagboxNW{#2}{#3}
33 }
```

Likewise for NE, SW, SE:

```
34 \newcommand{\LWR@diagboxTNE}[3]{%
35 \BlockClassSingle{diagboxtitleN}{#1}
36 \LWR@diagboxNE{#2}{#3}
37 }
38
39 \newcommand{\LWR@diagboxTSW}[3]{%
40 \LWR@diagboxSW{#2}{#3}
41 \BlockClassSingle{diagboxtitleS}{#1}
42 }
43
44 \newcommand{\LWR@diagboxTSE}[3]{%
45 \LWR@diagboxSE{#2}{#3}
46 \BlockClassSingle{diagboxtitleS}{#1}
47 }
```

```
\diagbox@triple  {\langle keys\rangle} {\langle A\rangle} {\langle T\rangle} {\langle B\rangle}

48 \def\diagbox@triple#1#2#3#4{%
49 \setkeys{diagbox}{dir=NW,#1}%
50 \csuse{LWR@diagboxT\diagbox@dir}{#3}{#2}{#4}%
51 }
```

File 64 **lwarp-draftwatermark.sty**

§ 156 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 65 **lwarp-easy-todo.sty**

§ 157 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 66 **lwarp-ebook.sty**

§ 158 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 67 **lwarp-ellipsis.sty**

§ 159 Package **ellipsis**

(Emulates or patches code by PETER J. HESLIN.)

Pkg ellipsis **ellipsis** is emulated.

```

1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}
4
5 \newcommand*{\midwordellipsis}{\,\textellipsis\,}

```

File 68 **lwarp-emptypage.sty**

§ 160 Package **emptypage**

Pkg emptypage **emptypage** is ignored.

for HTML output: Discard all options for **lwarp-emptypage**:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

File 69 **lwarp-endfloat.sty**

§ 161 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}{}
14 \def\efloattype{}
15 \providecommand\efloatheading[1]{}
16 \providecommand\efloatpreamble{}
17 \providecommand\efloatpostamble{}

```

File 70 **lwarp-endheads.sty**

§ 162 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 \titleinnotesttrue
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 71 **lwarp-endnotes.sty**

§ 163 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 72 **lwarp-enumerate.sty**

\$ 164 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 73 **lwarp-enumitem.sty**

\$ 165 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 74 **lwarp-epigraph.sty**

§ 166 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@print@mbox{text-align:right}}{ }\{epigraph\}
16 \qitem{#1}{#2}
17 \end{LWR@BlockClassWP}
18 }
19
20 \DeclareDocumentEnvironment{epigraphs}{}
21 {\LWR@BlockClassWP{\LWR@print@mbox{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 75 **lwarp-epstopdf.sty**

§ 167 Package **epstopdf**

Pkg epstopdf **epstopdf** is ignored.

Filenames should be used without a suffix so that SVG, PNG, or JPG versions of the file will be used for HTML output.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{epstopdf}

2 \providecommand*{\epstopdfsetup}[1]{}
3 \providecommand*{\epstopdfcall}[1]{}
4 \providecommand*{\epstopdfDeclareGraphicsRule}[4]{}

```

File 76 **lwarp-epstopdf-base.sty**

§ 168 Package **epstopdf-base**

Pkg epstopdf-base **epstopdf-base** is ignored.

Filenames should be used without a suffix so that SVG, PNG, or JPG versions of the file will be used for HTML output.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{epstopdf-base}[2016/05/15]

2 \providecommand*\epstopdfsetup[1]{}
3 \providecommand*\epstopdfcall[1]{}
4 \providecommand*\epstopdfDeclareGraphicsRule[4]{}

```

File 77 **lwarp-errata.sty**

§ 169 Package **errata**

(Emulates or patches code by MICHAEL KOHLHASE.)

Pkg errata **errata** is patched for use by **lwarp**.

This is for v0.3 of **errata**. A newer version of **errata** with more features is under development, at which time the **lwarp** version will have to be updated.

for HTML output: Macros are being defined with the math dollar, so enable the HTML version during package loading:

```

1 \StartDefiningMath

```

Now load the package:

```

2 \LWR@ProvidesPackagePass{errata}[2006/11/12]

```

Patches for dynamic inline math:

```

3 \xpatchcmd{\erratumAdd}
4   {$_a^{\arabic{erratum}}}$}
5 %   {\StartDynamicMath$_a^{\arabic{erratum}}$\StopDynamicMath}
6   {\textsubscript{a}\textsuperscript{\arabic{erratum}}}
7   {}

```

```

8      {
9          \PackageError{lwarp}
10             {Unable to patch the erratum package, erratumAdd macro}
11             {Please contact the author of the lwarp package.}
12      }
13
14 \xpatchcmd{\erratumDelete}
15     {$_d^{\arabic{erratum}}}$}
16 %     {\StartDynamicMath$_d^{\arabic{erratum}}$\StopDynamicMath}
17     {\textsubscript{d}\textsuperscript{\arabic{erratum}}}
18     {}
19     {
20         \PackageError{lwarp}
21             {Unable to patch the erratum package, erratumDelete macro}
22             {Please contact the author of the lwarp package.}
23     }
24
25 \xpatchcmd{\erratumReplace}
26     {$_r^{\arabic{erratum}}}$}
27 %     {\StartDynamicMath$_r^{\arabic{erratum}}$\StopDynamicMath}
28     {\textsubscript{r}\textsuperscript{\arabic{erratum}}}
29     {}
30     {
31         \PackageError{lwarp}
32             {Unable to patch the erratum package, erratumReplace macro}
33             {Please contact the author of the lwarp package.}
34     }
35
36 \xpatchcmd{\erratum}
37     {$_a$}
38 %     {\StartDynamicMath$_a$\StopDynamicMath}
39     {\textsubscript{a}}
40     {}
41     {
42         \PackageError{lwarp}
43             {Unable to patch the erratum package, eDelete macro}
44             {Please contact the author of the lwarp package.}
45     }
46
47 \xpatchcmd{\erratum}
48     {$_d^{\@thefnmark}}$}
49 %     {\StartDynamicMath$_d^{\@thefnmark}$\StopDynamicMath}
50     {\textsubscript{d}\textsuperscript{\@thefnmark}}
51     {}
52     {
53         \PackageError{lwarp}
54             {Unable to patch the erratum package, eDelete macro}
55             {Please contact the author of the lwarp package.}
56     }
57

```

```

58 \xpatchcmd{\erratum}
59   {$_r^{\@thefnmark}$}
60 %   {\StartDynamicMath$_r^{\@thefnmark}$\StopDynamicMath}
61   {\textsubscript{r}\textsuperscript{\@thefnmark}}
62   {}
63   {
64     \PackageError{lwarp}
65     {Unable to patch the erratum package, eReplace macro}
66     {Please contact the author of the lwarp package.}
67   }

```

Finish the current page's errata before closing and reloading the list:

```
68 \preto\PrintErrata{\LWR@orignewpage}
```

No longer defining math macros with the HTML \$:

```
69 \StopDefiningMath
```

File 78 **lwarp-eso-pic.sty**

§ 170 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 79 **lwarp-everypage.sty**

§ 171 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 80 **lwarp-everyshi.sty**

§ 172 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 81 **lwarp-extramarks.sty**

§ 173 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 82 **lwarp-fancybox.sty**

§ 174 Package **fancybox**

(Emulates or patches code by TIMOTHY VAN ZANDT.)

Pkg fancybox **fancybox** is supported with some patches.

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

```

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

```

[framing alternatives](#) \fbox works with **fancybox**. Also see **lwarp**'s \fboxBlock macro and fminipage environment for alternatives to \fbox for framing environments.

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

```

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

```

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

```

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

```

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

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

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

⚠ **sectioning or displaymath**

```

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

```

and likewise for equations or display math.

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

```

1 \LWR@ProvidesPackagePass{fancybox}

```

After the preamble is loaded, after any patches to Verbatim:

```

2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching fancybox.}

```

`\VerbatimFootnotes` Patched to use the new version.

```
4 \def\VerbatimFootnotes{%
5 \let\@footnotetext\V@footnotetext%
6 \let\LWR@footnotetext\V@footnotetext% lwarp
7 }
```

`V@@footnotetext` Patches in a subset of **lwarp's** `\LWR@footnotetext` to the **fancyvrb** version of `V@@footnotetext`.

```
8 \def\V@@footnotetext{%
9 \LWR@traceinfo{V@footnotetext}%
10 \global\setbox\LWR@footnotes=\vbox\bgroup%
```

Add to any current footnotes:

```
11 \unvbox\LWR@footnotes%
```

Remember the footnote number for `\ref`:

```
12 \protected@edef\@currentlabel{%
13 \csname p@footnote\endcsname\@thefnmark%
14 }% @currentlabel
```

Use HTML superscripts in the footnote even inside a `lateximage`:

```
15 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

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

```
16 \ifthenelse{%
17 \boolean{LWR@doingstartpars} \AND%
18 \cnttest{\value{LWR@lateximagedepth}}{=}{0}}%
19 }%
20 {}%
21 {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
22 \@makefntext{}%
23 \bgroup%
24 \aftergroup{V@@@footnotetext}%
25 \ignorespaces%
26 }%
```

```
27 }% AfterEndPreamble
```

```
28 \renewcommand*{\@shadowbox}[1]{%
29 \ifbool{FormatWP}%
30 {\InlineClass[border:1px solid black]{shadowbox}{#1}}%
31 {\InlineClass{shadowbox}{#1}}%
32 }
33
```

```

34 \renewcommand*{\@doublebox}[1]{%
35 \ifbool{FormatWP}%
36 {\InlineClass[border:1px double black]{doublebox}{#1}}%
37 {\InlineClass{doublebox}{#1}}%
38 }
39
40 \renewcommand*{\@ovalbox}[2]{%
41 \ifbool{FormatWP}%
42 {\InlineClass[border:1px solid black; border-radius:1ex]{ovalbox}{#2}}%
43 {%
44   \ifthenelse{\isequivalentto{#1}{\thinlines}}%
45   {\InlineClass{ovalbox}{#2}}%
46   {\InlineClass{Ovalbox}{#2}}%
47 }%
48 }

```

Convert minipages, parboxes, and lists into linear text using the LWR@nestspan environment:

```

49 \let\LWR@origSbox\Sbox
50
51 \def\Sbox{\LWR@origSbox\LWR@nestspan}
52
53
54 \let\LWR@origendSbox\endSbox
55
56 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

Beqnarray is adapted for MATHJAX or enclosed inside a lateximage:

```

57 \RenewEnviron{Beqnarray}
58 {\LWR@eqnarrayfactor}
59
60 \csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

\GenericCaption is enclosed in an HTML block:

```

61 \renewcommand{\GenericCaption}[1]{%
62 \LWR@figcaption%
63 #1%
64 \endLWR@figcaption%
65 }

```

Btrivlist is enclosed in an HTML block. This is a tabular, and does not use \item.

```

\trivlist   {\</c/r>} [<t/c/b>]
66 \RenewDocumentEnvironment{Btrivlist}{m o}

```

```

67 {%
68   \begin{BlockClass}{Btrivlist}%
69   \tabular{#1}%
70 }
71 {%
72   \endtabular%
73   \end{BlockClass}%
74 }

```

Btrivlist is also neutralized when used inside a span:

```

75 \AtBeginEnvironment{LWR@nestspan}{%
76 \RenewDocumentEnvironment{Btrivlist}{m o}{-}{-}%
77 }

```

lwarp's handling of \item is patched to accept fancybox's optional arguments:

```

78 \let\LWRFB@origitemizeitem\LWR@itemizeitem
79 \let\LWRFB@origdescitem\LWR@descitem
80
81 \RenewDocumentCommand{\LWR@itemizeitem}{d()o}{%
82   \IfValueTF{#2}{%
83     \LWRFB@origitemizeitem[#2]%
84   }{%
85     \LWRFB@origitemizeitem%
86   }%
87 }
88
89 \RenewDocumentCommand{\LWR@descitem}{d()o}{%
90   \IfValueTF{#2}{%
91     \LWRFB@origdescitem[#2]~%
92   }{%
93     \LWRFB@origdescitem%
94   }%
95 }
96
97 \RenewDocumentCommand{\LWR@nestspanitem}{d()}{%
98 \if@newlist\else{\LWR@htmltagc{br /}}\fi%
99 \LWR@origitem%
100 }

```

The various boxed lists become regular lists:

```

100 \renewenvironment{Bitemize}[1][]{\begin{itemize}}{\end{itemize}}
101 \renewenvironment{Benumerate}[1][]{\begin{enumerate}}{\end{enumerate}}
102 \renewenvironment{Bdescription}[1][]{\begin{description}}{\end{description}}

```

`\boxput` simply prints one then the other argument, side-by-side instead of above and behind:

```
103 \RenewDocumentCommand{\boxput}{s d() m m}{%
104 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
105 }
```

Neutralized commands:

```
106 \RenewDocumentCommand{\fancyput}{s d() m}{%
107 \RenewDocumentCommand{\thisfancyput}{s d() m}{%
108
109 \RenewDocumentCommand{\fancypage}{m m}{%
110 \RenewDocumentCommand{\thisfancypage}{m m}{%
111
112 \def\LandScape#1{%
113 \def\endLandScape{%
114 \def\@LandScape#1#2#3{%
115 \def\endLandScape{%
```

Low-level patches for `UseVerbatim` and friends:

```
116 \let\LWRFB@UseVerbatim\UseVerbatim
117 \renewcommand*{\UseVerbatim}[1]{%
118 \LWR@atbeginverbatim{3}{Verbatim}%
119 \LWRFB@UseVerbatim{#1}%
120 \LWR@afterendverbatim{.5}%
121 }
122
123 \let\LWRFB@LUseVerbatim\LUseVerbatim
124
125 \renewcommand*{\LUseVerbatim}[1]{%
126 \LWR@atbeginverbatim{3}{LVerbatim}%
127 \noindent%
128 \LWRFB@LUseVerbatim{#1}%
129 \LWR@afterendverbatim{.5}%
130 }
131
132 \def\@BUseVerbatim[#1]#2{%
133 \LWR@atbeginverbatim{3}{BVerbatim}%
134 \LWRFB@UseVerbatim{#2}%
135 \LWR@afterendverbatim{.5}%
136 }
```

File 83 **lwarp-fancyheadings.sty**

§ 175 Package **fancyheadings**

Pkg fancyheadings **fancyheadings** is superceded by **fancyhdr**.

for HTML output: 1 \LWR@loadnever{fancyheadings}{fancyhdr}

File 84 **lwarp-fancyhdr.sty**

§ 176 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 85 **lwarp-fancyref.sty**

§ 177 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 86 **lwarp-fancyvrb.sty**

§ 178 Package **fancyvrb**

(Emulates or patches code by TIMOTHY VAN ZANDT.)

Pkg fancyvrb **fancyvrb** is supported with some patches.

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

```

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

```

and likewise for equations or display math.

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

```

1 \RequirePackage{xcolor}% for \convertcolorspec
2
3 \LWR@ProvidesPackagePass{fancyvrb}

```

Initial default patch for fancyvrb:

```

4 \fvset{frame=none}%

```

After the preamble is loaded, after any patches to Verbatim:

```

5 \AfterEndPreamble{
6 \LWR@traceinfo{Patching fancyvrb.}

```

\VerbatimFootnotes Patched to use the new version.

```

7 \def\VerbatimFootnotes{%
8 \let\@footnotetext\V@footnotetext%
9 \let\footnote\V@footnote%
10 \let\LWR@footnotetext\V@footnotetext% lwarp
11 }

```

`\V@@footnotetext` Patches in a subset of **lwarp's** `\LWR@footnotetext` to the **fancyvrb** version of `\V@@footnotetext`.

```
12 \def\V@@footnotetext{%
13 \LWR@traceinfo{V@footnotetext}%
14 \global\setbox\LWR@footnotes=\vbox\bgroup%
```

Add to any current footnotes:

```
15 \unvbox\LWR@footnotes%
```

Remember the footnote number for `\ref`:

```
16 \protected@edef\@currentlabel{%
17 \csname p@footnote\endcsname\@thefnmark%
18 }% \@currentlabel
```

Use HTML superscripts in the footnote even inside a `lateximage`:

```
19 \renewrobustcmd{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

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

```
20 \ifthenelse{%
21 \boolean{LWR@doingstartpars} \AND%
22 \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
23 }%
24 {}%
25 {\LWR@htmltagc{\LWR@tagregularparagraph}}%
```

Append the footnote to the list:

```
26 \@makefntext{}%

27 \bgroup%
28 \aftergroup{\V@@@footnotetext}%
29 \ignorespaces%
30 }%

31 \preto\FVB@Verbatim{\LWR@forcenewpage}
32 \preto\FVB@LVerbatim{\LWR@forcenewpage}
33 % \preto\FVB@BVerbatim{\LWR@forcenewpage}% Fails, so done below.
```

Simplified to remove PDF formatting:

```
34 \def\FV@BeginListFrame@Single{%
35 \FV@SingleFrameLine{z}%
36 }
37
38 \def\FV@EndListFrame@Single{%
39 \FV@SingleFrameLine{@ne}%
40 }
41
42 \def\FV@BeginListFrame@Lines{%
```

```

43 \FV@SingleFrameLine{\z}%
44 }
45
46 \def\FV@EndListFrame@Lines{%
47   \FV@SingleFrameLine{\@ne}%
48 }
49
50 \renewcommand*\FV@SingleFrameSep{}

```

Adds HTML formatting:

```

51 \def\FV@BUseVerbatim#1{%
52   \LWR@atbeginverbatim[\LWR@FVstyle]{0}{verbatim}%
53   \FV@BVerbatimBegin#1\FV@BVerbatimEnd%
54   \LWR@afterendverbatim{0}%
55 }

```

`\LWR@FVstyle` Holds the style of the verbatim.

```

56 \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.

```

57 \newcommand*\LWR@fvstartnone{%
58 \LWR@traceinfo{fvstartnone}%
59 % \hbox to\z@{
60 \LWR@atbeginverbatim[\LWR@FVstyle]{0}{verbatim}%
61 % }%
62 }
63
64 \newcommand*\LWR@fvendnone{%
65 \LWR@traceinfo{fvendnone}%
66 % \hbox to\z@{
67 \LWR@afterendverbatim{0}%
68 % }%
69 }
70
71 \newcommand*\LWR@fvstartsingle{%
72 \LWR@traceinfo{fvstartsingle}%
73 \LWR@fvstartnone%
74 \FV@BeginListFrame@Single%
75 }
76
77 \newcommand*\LWR@fvendsingle{%
78 \LWR@traceinfo{fvendsingle}%
79 \FV@EndListFrame@Single%

```

```

80 \LWR@fvendnone%
81 }
82
83 \newcommand*{\LWR@fvstartline}{%
84 \LWR@traceinfo{fvstartline}%
85 \LWR@fvstartnone%
86 % \setlength{\LWR@templengthone}{\baselineskip}%
87 \FV@BeginListFrame@Lines%
88 % \setlength{\baselineskip}{\LWR@templengthone}%
89 % \setlength{\baselineskip}{5pt}%
90 }
91
92 \newcommand*{\LWR@fvendline}{%
93 \LWR@traceinfo{fvendline}%
94 \FV@EndListFrame@Lines%
95 \LWR@fvendnone%
96 }

```

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the **fancyvrb** package.

```

97 \newcommand*{\LWR@FVfindbordercolor}{%
98 \FancyVerbRuleColor%
99 \LWR@findcurrenttextcolor%
100 \color{black}%
101 }
102
103 % border width of \FV@FrameRule
104 \newcommand*{\LWR@FVborderstyle}[1]{%
105 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ; %
106 \LWR@FVfindbordercolor %
107 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt %
108 solid \LWR@origpound\LWR@tempcolor ; %
109 }
110
111 \def\FV@Frame@none{%
112 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle}%
113 \let\FV@BeginListFrame\LWR@fvstartnone%
114 \let\FV@LeftListFrame\relax%
115 \let\FV@RightListFrame\relax%
116 \let\FV@EndListFrame\LWR@fvendnone%
117
118 \FV@Frame@none% default values
119
120 \def\FV@Frame@single{%
121 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{}}%
122 \let\FV@BeginListFrame\LWR@fvstartsingle%
123 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
124 \let\FV@RightListFrame\FV@RightListFrame@Single%

```

```

125 \let\FV@EndListFrame\LWR@fvendsingle}
126
127 \def\FV@Frame@lines{%
128 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}}%
129 }%
130 }%
131 \let\FV@BeginListFrame\LWR@fvstartline%
132 \let\FV@LeftListFrame\relax%
133 \let\FV@RightListFrame\relax%
134 \let\FV@EndListFrame\LWR@fvendline}
135
136 \def\FV@Frame@topline{%
137 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-top}}%
138 \let\FV@BeginListFrame\LWR@fvstartline%
139 \let\FV@LeftListFrame\relax%
140 \let\FV@RightListFrame\relax%
141 \let\FV@EndListFrame\LWR@fvendnone}
142
143 \def\FV@Frame@bottomline{%
144 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-bottom}}%
145 \let\FV@BeginListFrame\LWR@fvstartnone%
146 \let\FV@LeftListFrame\relax%
147 \let\FV@RightListFrame\relax%
148 \let\FV@EndListFrame\LWR@fvendline}
149
150 \def\FV@Frame@leftline{%
151 \renewcommand*{\LWR@FVstyle}{\LWR@currenttextcolorstyle\LWR@FVborderstyle{-left}}%
152 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
153 \ifx\FancyVerbFillColor\relax%
154 \let\FV@FrameFillLine\relax%
155 \else%
156 \@tempdima\FV@FrameRule\relax%
157 \multiply\@tempdima-\tw@%
158 \edef\FV@FrameFillLine{%
159 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}}%
160 \kern-\number\@tempdima sp}}%
161 \fi%
162 \let\FV@BeginListFrame\LWR@fvstartnone%
163 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
164 \let\FV@RightListFrame\relax%
165 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the **fancyvrb** package.

```

166 \def\FV@SingleFrameLine#1{%
167 % \hbox to\z@{%
168 % \kern\leftmargin
169 % \ifnum#1=\z@\relax

```

```

170     \let\FV@Label\FV@LabelBegin
171   \else
172     \let\FV@Label\FV@LabelEnd
173   \fi
174   \ifx\FV@Label\relax
175 %     \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
176   \else
177     \ifnum#1=\z@
178 %     \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
179     \ifx\FV@LabelPositionTopLine\relax
180     \else
181     \LWR@FVfindbordercolor
182     \LWR@htmltagc{%
183       div class="fancyvrblabel" % extra space
184       style="color: \LWR@origpound\LWR@tempcolor"%
185     }
186     \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
187     \LWR@htmltagc{/div}
188   \fi
189   \else
190 %     \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
191     \ifx\FV@LabelPositionBottomLine\relax
192     \else
193     \LWR@FVfindbordercolor
194
195     \LWR@htmltagc{%
196       div class="fancyvrblabel" % extra space
197       style="color: \LWR@origpound\LWR@tempcolor"%
198     }
199     \LWR@origtextrm{\FV@LabelEnd}
200     \LWR@htmltagc{/div}
201   \fi
202   \fi
203   \fi
204 %   \hss
205 %   }
206 }

```

Processes each line, adding optional line numbers. Original code is from the **fancyvrb** package.

```

207 \def\FV@ListProcessLine#1{%
208   \hbox to \hsize{%
209 %     \kern\leftmargin
210     \hbox to \VerbatimHTMLWidth {%
211       \ifcvoid\FV@LeftListNumber}{-}{\kern 2.5em}%
212       \FV@LeftListNumber%
213 %     \FV@LeftListFrame
214     \FancyVerbFormatLine{#1}%

```

```

215      \hss%
216 %      \FV@RightListFrame
217      \FV@RightListNumber%
218    }%
219      \hss% required to avoid underfull hboxes
220 }
221 }

```

Env **BVerbatim**

```

222 \AtBeginEnvironment{BVerbatim}
223 {%
224 \LWR@forcenewpage% instead of \preto
225 \LWR@atbeginverbatim{0}{bverbatim}%
226 }
227
228 \AfterEndEnvironment{BVerbatim}
229 {%
230 \LWR@afterendverbatim{0}%
231 }

```

End of the modifications to make at the end of the preamble:

```

232 } % \AfterEndPreamble

```

File 87 **lwarp-figcaps.sty**

§ 179 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 88 **lwarp-figsize.sty**

§ 180 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 89 **lwarp-fix2col.sty**

§ 181 Package **fix2col**

Pkg **fix2col** **fix2col** is ignored.


for HTML output: 1 \LWR@ProvidesPackageDrop{fix2col}

File 90 **lwarp-fixme.sty**

§ 182 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 }
```

```

33 }
34
35
36 \def\FXFaceTargetHTMLStyle{font-style:italic}
37
38 \renewcommand\FXTargetLayoutPlain[2]{%
39   \InlineClass[\FXFaceTargetHTMLStyle]{fixmetarget}{#2}%
40 }

```

File 91 **lwarp-fixmetodonotes.sty**

§ 183 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 92 **lwarp-flafter.sty**

§ 184 Package **flafter**

Pkg flafter **flafter** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{flafter}
2 \providecommand\fl@trace[1]{

```

File 93 **lwarp-float.sty**

§ 185 Package **float**

(Emulates or patches code by ANSELM LINGNAU.)

Pkg float **float** is emulated.

Float styles boxed and ruled are emulated by css and a float class according to style.

The HTML <figure> class is set to the float type, so css may also be used to format the float and its caption, according to float type. Furthermore, an additional class is set to the float style: plain, plaintop, boxed, or ruled, so css may be used to format by float style as well. Default formatting by css is provided for ruled and boxed styles.

for HTML output: 1 \LWR@ProvidesPackageDrop{float}[2001/11/08]

\listof See section 71.2 for the \listof command.

\LWR@floatstyle The default float style:

2 \newcommand*{\LWR@floatstyle}{plain}

\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.

```
3 \NewDocumentCommand{\newfloat}{m m m o}{%
4 \IfValueTF{#4}{%
5 {\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}%
6 {\DeclareFloatingEnvironment[fileext=#3]{#1}}}%

```

Remember the float style:

```
7 \csedef{\LWR@floatstyle@#1}{\LWR@floatstyle}%

```

newfloat package automatically creates the \listof command for new floats, but **float** does not, so remove \listof here in case it is manually created later.

```
8 \cslet{\listof#1s}\relax%
9 \cslet{\listof#1es}\relax%

```

Likesize, **newfloat** also creates `\l@<type>`, but **float** does not, so remove it here:

```
10 \cslet{l@#1}\relax%
11 }
```

`\floatname` `{<type>} {<name>}`

Sets the text name of the float, such as “Figure”.

```
12 \NewDocumentCommand{\floatname}{m +m}{%
13   \SetupFloatingEnvironment{#1}{name=#2}%
14 }
```

`\floatplacement` `{<type>} {<placement>}`

Float placement is ignored.

```
15 \newcommand*{\floatplacement}[2]{%
16   \SetupFloatingEnvironment{#1}{placement=#2}%
17 }
```

`\floatstyle` `{<style>}`

Remember the style for future floats:

```
18 \newcommand{\floatstyle}[1]{%
19   \def\LWR@floatstyle{#1}%
20 }
```

`\restylefloat` `* {<type>}`

Remember the style for this float:

```
21 \NewDocumentCommand{\restylefloat}{s m}{%
22   \csedef{LWR@floatstyle@#2}{\LWR@floatstyle}%
23 }
```

File 94 **lwarp-floatflt.sty**

§ 186 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 `[< >]` `offset {<type>} {<width>}` 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}; % extra space
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 95 **lwarp-floatpag.sty**

§ 187 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 96 **lwarp-floatrow.sty**

§ 188 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 \StopDefiningTabulars before and after defining macros using \ttabbox with a tabular inside. See section 9.9.

⚠ subfig package When combined with the **subfig** package, while inside a subfloatrow \ffigbox and \ttabbox must have the caption in the first of the two of the mandatory arguments.

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

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

After everything has loaded, remember whether **subcaption** was loaded. If not, it is assumed that **subfig** is used instead:

```
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    [<width>] [<height>] [<vposn>] {<caption commands>} {<contents>}
68 \newfloatcommand{\ttabbox}{table}[\captop][\FBwidth]
```

```
\fcapside   [<width>] [<height>] [<vposn>] {<caption commands>} {<contents>}
69 \newfloatcommand{\fcapside}{figure}[\capbeside] []
```

Env floatrow [<numfloats>]

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 97 **lwarp-fltrace.sty**

§ 189 Package **fltrace**

Pkg fltrace **fltrace** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fltrace}

2 \def\tracefloats{}
3 \def\tracefloatsoff{}
4 \def\tracefloatvals{}
```

File 98 **lwarp-flushend.sty**

§ 190 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{macrocode}
3 %
4 %      \begin{macrocode}
5 \newcommand*{\flushend}{}
6 \newcommand*{\raggedend}{}
7 \newcommand*{\flushcolsend}{}
8 \newcommand*{\raggedcolsend}{}
9 \newcommand*{\atColsBreak}[1]{}
10 \newcommand*{\atColsEnd}[1]{}
11 \newcommand*{\showcolsendrule}{}

```

File 99 **lwarp-fncychap.sty**

§ 191 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 100 **lwarp-fnlineno.sty**

§ 192 Package **fnlineno**

Pkg fnlineno **fnlineno** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{fnlineno}

File 101 **lwarp-fnpos.sty**

§ 193 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 102 **lwarp-fontenc.sty**

§ 194 Package **fontenc**

Pkg fontenc If using pdf_{TEX}, **lwarp** used to require **fontenc** be loaded before **lwarp**, but now **lwarp** itself loads `\fontenc` with T1 encoding, which **lwarp** requires. **fontenc** is now allowed to be loaded with another encoding after **lwarp**.

lwarp-fontenc is no longer necessary, but is still provided to overwrite older versions.

for HTML output: 1 \LWR@ProvidesPackagePass{fontenc}

File 103 **lwarp-fontspec.sty**

§ 195 Package **fontspec**

Pkg fontspec Error if **fontspec** is loaded after **lwarp**.

Discard all options for **lwarp-fontspec**:

for HTML output: 1 \LWR@ProvidesPackageDrop{fontspec}[2017/11/09]

2 \LWR@loadbefore{fontspec}

File 104 **lwarp-footmisc.sty**

§ 196 Package **footmisc**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg footmisc **footmisc** is emulated.

lwarp incidentally happens to emulate the `stable` option.

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 105 **lwarp-footnote.sty**

§ 197 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 106 **lwarp-footnotehyper.sty**

§ 198 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 107 **lwarp-footnpag.sty**

§ 199 Package **footnpag**

Pkg footnpag **footnpag** is ignored.


for HTML output: 1 \LWR@ProvidesPackageDrop{footnpag}

File 108 **lwarp-forest.sty**

§ 200 Package **forest**

(Emulates or patches code by SAŠO ŽIVANOVIĆ.)

Pkg forest **forest** is patched for use by **lwarp**.

 **\Forest*** The starred version of the macro **\Forest*** is not supported. **lwarp** encases each **lateximage** in an environment, so the global results of the starred **\Forest*** are lost.

for HTML output: 1 \LWR@ProvidesPackagePass{forest}

```
2 \BeforeBeginEnvironment{forest}{\begin{lateximage}[forest]}
3
4 \AfterEndEnvironment{forest}{\end{lateximage}}
5
6 \RenewDocumentCommand{\Forest}{s D(){} m}{%
```

```

7 \forest@config{#2}%
8 \IfBooleanTF{#1}{%
9     \PackageError{lwarp-forest}%
10    {Starred \Forest is not supported}%
11    {Lwarp uses an environment for images, but \Forest* cannot work in an environment.}%
12    \let\forest@next\forest@env%
13    }{\let\forest@next\forest@group@env}%
14    \begin{lateximage}[forest]%      lwarp
15 \forest@next{#3}%
16 \end{lateximage}%                  lwarp
17 }
```

File 109 **lwarp-framed.sty**

§ 201 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@origpound\LWR@tempcolor]{shaded}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
```

```

25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \LWR@origpound\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@origpound\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47
48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \LWR@origpound\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@origpound\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{\% not empty
67   \begin{BlockClass}[background: \LWR@origpound\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@origpound\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 110 **lwarp-ftnright.sty**

§ 202 Package **ftnright**

Pkg ftnright **ftnright** is ignored.

for HTML output: Discard all options for **lwarp-ftnright**:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

File 111 **lwarp-fullpage.sty**

§ 203 Package **fullpage**

Pkg fullpage **fullpage** is ignored.

for HTML output: Discard all options for **lwarp-fullpage**:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

File 112 **lwarp-fullwidth.sty**

§ 204 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 113 **lwarp-fwlw.sty**

§ 205 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@fwlwhead{}
6 \def\ps@NextWordFoot{}

```

File 114 **lwarp-geometry.sty**

§ 206 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 115 **lwarp-glossaries.sty**

§ 207 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 xindyLanguage	The lwarp package option <code>xindyLanguage</code> sets the language used by xindy . This is passed to xindy using its <code>-L</code> option, and is used for both index and glossary generation.
Default: english	
Opt xindyCodepage	The option <code>xindycodepage</code> sets the codepage used by xindy . This is passed to xindy using its <code>-C</code> option, and is used for both index and glossary generation.
Default: utf8	
Opt lwarpmk printglossary	lwarpmk has the commands <code>lwarpmk printglossary</code> and <code>lwarpmk htmlglossary</code> to process the glossaries created by glossaries using xindy .
Opt lwarpmk htmlglossary	
for HTML output:	<pre> 1 \PassOptionsToPackage{xindy}{glossaries} 2 \LWR@ProvidesPackagePass{glossaries} 3 \setupglossaries{nonumberlist} 4 \setglossarystyle{index} </pre>

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, which could be a different HTML file. 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 }

```

lwarp's sectioning commands cannot handle robust macros when splitting HTML into named filenames. **glossaries** uses `\translate` in sectioning names, and `\translate` is robust and cannot be expanded. The following pre-expands the translations at this moment, making use of `\translatelet`.

```

18 \newcommand*{\LWR@comp@glossaryname}{\translate{Glossary}}
19
20 \ifdefstrequal{\glossaryname}{\LWR@comp@glossaryname}{
21   \translatelet\LWR@translatetemp{Glossary}
22   \edef\glossaryname{\LWR@translatetemp}
23 }{}
24
25 \newcommand*{\LWR@comp@acronymname}{\translate{Acronym}}
26
27 \ifdefstrequal{\acronymname}{\LWR@comp@acronymname}{
28   \translatelet\LWR@translatetemp{Acronym}
29   \edef\acronymname{\LWR@translatetemp}
30 }{}
31
32 \newcommand*{\LWR@comp@glssymbolsgroupname}{\translate{Symbols (glossaries)}}
33
34 \ifdefstrequal{\glssymbolsgroupname}{\LWR@comp@glssymbolsgroupname}{
35   \translatelet\LWR@translatetemp{Symbols (glossaries)}
36   \edef\glssymbolsgroupname{\LWR@translatetemp}
37 }{}
38
39 \newcommand*{\LWR@comp@glslnumbersgroupname}{\translate{Numbers (glossaries)}}
40
41 \ifdefstrequal{\glslnumbersgroupname}{\LWR@comp@glslnumbersgroupname}{
42   \translatelet\LWR@translatetemp{Numbers (glossaries)}
43   \edef\glslnumbersgroupname{\LWR@translatetemp}
44 }{}

```

File 116 **lwarp-glossary.sty**

§ 208 Package **glossary**

Pkg **glossary** **glossary** is superceded by **glossaries**.

for HTML output: 1 \LWR@loadnever{glossary}{glossaries}

File 117 **lwarp-graphics.sty**

§ 209 Package **graphics**

(Emulates or patches code by D. P. CARLISLE.)

Pkg graphics **graphics** is emulated.

for HTML output: 1 \LWR@ProvidesPackagePass{graphics}

§ 209.1 **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 }
```

§ 209.2 **Length conversions and graphics options**

 **whitespace**

A scaled image in \TeX by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

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} % extra space
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}[c]{%
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}[true]{}
76 \define@key{igraph}{viewport}{}
77 \define@key{igraph}{trim}{}
78 \define@key{igraph}{totalheight}{}
79 \define@key{igraph}{keepaspectratio}[true]{}
80 \define@key{igraph}{clip}[true]{}
81 \define@key{igraph}{draft}[true]{}
82 \define@key{igraph}{type}{}
83 \define@key{igraph}{ext}{}
84 \define@key{igraph}{read}{}
85 \define@key{igraph}{command}{}

```

New in v1.1a:

```

86 \define@key{igraph}{quite}{}
87 \define@key{igraph}{page}{}
88 \define@key{igraph}{pagebox}{}
89 \define@key{igraph}{interpolate}[true]{}

```

New in v1.1b:

```

90 \define@key{igraph}{decodearray}{}

```

§ 209.3 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.

```

91 \newcommand*\LWR@rotstyle[2]{%
92   #1transform:rotate(-#2deg);
93 }

```

`\LWR@scalestyle` $\{\langle prefix \rangle\} \{\langle xscale \rangle\} \{\langle yscale \rangle\}$

Prints the scale style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:scale` style.

```

94 \newcommand*\LWR@scalestyle[3]{%
95   #1transform:scale(#2,#3);
96 }

```

§ 209.4 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
97 \newbool{LWR@infloatrow}
98 \boolfalse{LWR@infloatrow}
```

`\LWR@opacity` For HTML, used only for `\includegraphics`.

`\LWR@opacity` may be set by the **transparent** package.

```
99 \def\LWR@opacity{1}
```

`\LWR@imagesizebox` Used to determine the actual image size if needed.

```
100 \newsavebox{\LWR@imagesizebox}
```

`\Gin@setfile` Sets the parsed filename.

```
101 \let\LWR@origGin@setfile\Gin@setfile
```

Key `Gin` `class` CSS class for the image.

Define the new class key for the print-mode version of `\includegraphics`, which is enabled inside a `lateximage`.

```
102 \AtBeginDocument{
103 \define@key{Gin}{class}{}
104 }
```

`\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.

```
105 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
106 {%
107 \LWR@traceinfo{LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
108 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```

109 \begingroup%
110 \ifthenelse{\cnttest{\value{LWR@minipagedepth}}{=}{0}}%
111 {%
112   \ifbool{LWR@infloatrow}%
113   {}
114   {% not in a minipage or a floatrow:
115     \setlength{\linewidth}{6in}%
116     \setlength{\textwidth}{6in}%
117     \setlength{\textheight}{9in}%
118   }%
119 }{}%

120 \begingroup%
121 \renewcommand*{\Gin@setfile}[3]{%
122 \LWR@traceinfo{Gin@setfile ##3}%
123 \xdef\LWR@parsedfilename{##3}%
124 }%
125 \Gininclude@graphics{\detokenize\expandafter{#4}}%
126 \endgroup%
127 \filename@parse{\LWR@parsedfilename}%
128 \LWR@traceinfo{\LWR@parsedfilename is \LWR@parsedfilename}%
129 % \LWR@sanitize{\LWR@parsedfilename}%

```

For correct em sizing during the width and height conversions:

```

130 \large%

```

Reset some defaults, possibly will be changed below if options were given:

```

131 \setlength{\LWR@igwidth}{0pt}%
132 \setlength{\LWR@igheight}{0pt}%
133 \renewcommand*{\LWR@igwidthstyle}{}%
134 \renewcommand*{\LWR@igheightstyle}{}%
135 \renewcommand*{\LWR@igorigin}{}%
136 \renewcommand*{\LWR@igangle}{}%
137 \renewcommand*{\LWR@igxscale}{1}%
138 \renewcommand*{\LWR@igyscale}{1}%
139 \renewcommand*{\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus **graphicx** syntax:

```

140 \IfValueF{#3}{%
141 \IfValueTF{#2}%
142 {\setkeys{igraph}{#2}}%
143 {\setkeys{igraph}{}}%
144 }%

```

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:

```

145 \ifbool{FormatWP}{%
146   \begingroup%
147   \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
148   \define@key{Gin}{angle}{}%
149   \IfBooleanTF{#1}%
150   {% starred
151     \IfValueTF{#3}%
152     {%
153       \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] [#3] {#4}}%
154     }%
155     {%
156       \IfValueTF{#2}%
157       {%
158         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* [#2] {#4}}%
159       }{%
160         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics* {#4}}%
161       }%
162     }%
163   }% starred
164   {% not starred
165     \IfValueTF{#3}%
166     {%
167       \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics [#2] [#3] {#4}}%
168     }%
169     {%
170       \IfValueTF{#2}%
171       {%
172         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics [#2] {#4}}%
173       }{%
174         \global\sbox{\LWR@imagesizebox}{\LWR@originincludegraphics {#4}}%
175       }%
176     }%
177   }% not starred
178   \endgroup%
179   \settowidth{\LWR@igwidth}{\usebox{\LWR@imagesizebox}}%
180   \global\renewcommand*{\LWR@igwidthstyle}{width:\LWR@printlength{\LWR@igwidth}}%
181   \settoheight{\LWR@igheight}{\usebox{\LWR@imagesizebox}}%
182   \global\renewcommand*{\LWR@igheightstyle}{height:\LWR@printlength{\LWR@igheight}}%
183 }{}%

```

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.

```

184 \LWR@traceinfo{\LWR@includegraphicsb: about to create href}%
185 \href{\LWR@parsedfilename}%
186 {% start of href
187 \LWR@traceinfo{\LWR@includegraphicsb: about to LWR@htmltag}%
188 \LWR@htmltag{% start of image tags
189 img src="%
190 \begingroup\@sanitize\LWR@parsedfilename\endgroup%
191 " \LWR@orignewline%

```

Only include a style tag if a width, height, angle, or scale was given:

```

192 \ifthenelse{
193   \NOT\equal{\LWR@igwidthstyle}{ } \OR
194   \NOT\equal{\LWR@igheightstyle}{ } \OR
195   \NOT\equal{\LWR@igorigin}{ } \OR
196   \NOT\equal{\LWR@igangle}{ } \OR
197   \NOT\equal{\LWR@igxscale}{1} \OR
198   \NOT\equal{\LWR@igyscale}{1}
199 }%
200 {\LWR@origtilde{ } style="%
201 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
202 {\LWR@igwidthstyle; }}%
203 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
204 {\LWR@igheightstyle; }}%
205 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
206 {\LWR@origtilde{ } transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@orignewline}{ }%
207 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
208 {%
209 \LWR@rotstyle{-ms-}{\LWR@igangle} % extra space
210 \LWR@rotstyle{-webkit-}{\LWR@igangle} % extra space
211 \LWR@rotstyle{}{\LWR@igangle} %
212 } }%
213 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
214 \NOT\equal{\LWR@igyscale}{1}}%
215 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale} % extra space
216 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale} % extra space
217 \LWR@scalestyle{}{\LWR@igxscale}{\LWR@igyscale}} % extra space
218 %
219 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
220 {opacity:\LWR@opacity; }%
221 }%
222 %
223 " \LWR@orignewline}{ }%

```

Set the class:

```

224 \LWR@origtilde{ } class="\LWR@igclass" \LWR@orignewline%
225 }% end of image tags
226 }% end of href

```

Return to original page size and font size:

```
227 \endgroup
228 \LWR@traceinfo{LWR@includegraphicsb done}%
229 }
```

`\includegraphics` [*<key=val>*] {*<filename>*}

Handles width and height, converted to fixed width and heights.

The user should always use no file suffix in the document source.

```
230 \AtBeginDocument{
231
232 \LWR@traceinfo{Patching includegraphics.}
233
234 \LetLtxMacro\LWR@originincludegraphics\includegraphics
235
236 \renewcommand*{\includegraphics}
237 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
238 \LWR@traceinfo{includegraphics}%
239 \LWR@ensuredoingapar%
240 \LWR@includegraphicsb%
241 }% includegraphics
242 }% AtBeginDocument
```

§ 209.5 Boxes

`\LWR@rotboxorigin` Holds the origin key letters.

```
243 \newcommand*{\LWR@rotboxorigin}{}
```

`\LWR@originname` {*<letter>*}

Given one \TeX origin key value, translate into an HTML origin word:

```
244 \newcommand*{\LWR@originname}[1]{%
245 \ifthenelse{\equal{#1}{t}}{top}{}%
246 \ifthenelse{\equal{#1}{b}}{bottom}{}%
247 \ifthenelse{\equal{#1}{c}}{center}{}%
248 \ifthenelse{\equal{#1}{l}}{left}{}%
249 \ifthenelse{\equal{#1}{r}}{right}{}%
250 }
```

`\LWR@originnames` $\{\langle letters \rangle\}$

Given one- or two-letter \TeX origin key values, translate into HTML origin words:

```
251 \newcommand*{\LWR@originnames}[1]{%
252 \StrChar{#1}{1}[\LWR@strresult]%
253 \LWR@originname{\LWR@strresult}
254 \StrChar{#1}{2}[\LWR@strresult]%
255 \LWR@originname{\LWR@strresult}
256 }
```

Handle the origin key for `\rotatebox`:

```
257 \define@key{krotbox}{origin}{%
258 \renewcommand*{\LWR@rotboxorigin}{#1}%
259 }
```

These keys are ignored:

```
260 \define@key{krotbox}{x}{}
261 \define@key{krotbox}{y}{}
262 \define@key{krotbox}{units}{}

```

`\rotatebox` $[\langle keyval list \rangle] \{\langle angle \rangle\} \{\langle text \rangle\}$

```
263 \AtBeginDocument{
```

The HTML version:

```
264 \NewDocumentCommand{\LWR@HTML@rotatebox}{0{} m +m}{%
```

Reset the origin to “none-given”:

```
265 \renewcommand*{\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
266 \setkeys{krotbox}{#1}%

```

Select inline-block so that HTML will transform this span:

```
267 \LWR@htmltagc{span style="display: inline-block; %
```

If an origin was given, translate and print the origin information:

```
268 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{} }{%
269 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```
270 \LWR@rotstyle{-ms-}{#2} % extra space
271 \LWR@rotstyle{-webkit-}{#2} % extra space
272 \LWR@rotstyle{}{#2} % extra space
273 "}\LWR@orignewline%
```

Print the text to be rotated:

```
274 \begin{LWR@nestspan}%
275 #3%
```

Close the span:

```
276 \LWR@htmltagc{/span}%
277 \end{LWR@nestspan}%
278 }
```

The high-level interface:

```
279 \LWR@formatted{rotatebox}
280
281 }% AtBeginDocument
```

`\scalebox` $\{\langle h\text{-scale}\rangle\} [\langle v\text{-scale}\rangle] \{\langle \text{text}\rangle\}$

```
282 \AtBeginDocument{
```

The HTML version:

```
283 \NewDocumentCommand{\LWR@HTML@scalebox}{m o m}{%
```

Select inline-block so that HTML will transform this span:

```
284 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
285 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
286 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
287 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} % extra space
288 "}%
```

Print the text to be scaled:

```
289 \begin{LWR@nestspan}%
290 #3%
```

Close the span:

```
291 \LWR@htmltagc{/span}%
292 \end{LWR@nestspan}%
293 }
```

The high-level interface:

```
294 \LWR@formatted{scalebox}
295
296 }% AtBeginDocument
```

`\reflectbox` $\{\langle text \rangle\}$

```
297 \AtBeginDocument{
298
299 \newcommand{\LWR@HTML@reflectbox}[1]{%
300 \scalebox{-1}[1]{#1}%
301 }% \reflectbox
302
303 \LWR@formatted{reflectbox}
304
305 }% AtBeginDocument
```

`\resizebox` $\{\langle h-length \rangle\} \{\langle v-length \rangle\} \{\langle text \rangle\}$

Simply prints its text argument.

```
306 \AtBeginDocument{
307
308 \NewDocumentCommand{\LWR@HTML@resizebox}{s m m m}{%
309 #4%
310 }
311
312 \LWR@formatted{resizebox}
313
314 }% AtBeginDocument
```

File 118 **lwarp-graphicx.sty**

§ 210 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}[2017/06/01]`

File 119 **lwarp-grffile.sty**

§ 211 Package **grffile**

 **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.

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 120 **lwarp-grid.sty**

§ 212 Package **grid**

`Pkg grid` **grid** is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{grid}`

`2 \newenvironment*{gridenv}{}{}`

File 121 **lwarp-gridset.sty**

§ 213 Package **gridset**

`Pkg gridset` **gridset** is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{gridset}`

```

2 \newcommand*\gridbase{}
3 \newcommand*\gridinterval{}
4 \newcommand*\savepos}[1]{}
5 \newcounter{gridcnt}
6 \newcommand*\vskipnextgrid{}
7 \newcommand*\thegridinfo}[1]{}
8 \newcommand*\theposinfo}[1]{}
9 \newcommand*\theypos}[1]{}

```

File 122 **lwarp-hang.sty**

§ 214 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{\hangleleftmargin}
5 \setlength{\hangleleftmargin}{0em}
6
7 \newcommand*\LWR@findhangleleftmargin{%
8 \setlength{\LWR@templengthone}{\hangleleftmargin}%
9 \addtolength{\LWR@templengthone}{\hangingindent}%
10 }
11
12 \newenvironment{hangingpar}
13 {
14   \LWR@findhangleleftmargin%
15   \BlockClass[%
16     \LWR@print@mbx{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
17     \LWR@print@mbx{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@findhangleleftmargin%
28     ul style="%
29     \LWR@print@mbx{list-style-type:none;} % extra space

```

```

30      \LWR@print@mbbox{%
31          margin-left:\LWR@printlength{\LWR@templengthone}%
32      } ; % extra space
33      \LWR@print@mbbox{%
34          text-indent:-\LWR@printlength{\hangingindent}%
35      }%
36      "%
37  }%
38  \let\item\LWR@itemizeitem%
39  \list{}{}%
40 }
41 {\endlist}
42
43 \newenvironment{compacthang}
44 {\hanginglist}
45 {\endhanginglist}
46
47 \newlength{\labeledleftmargin}
48 \setlength{\labeledleftmargin}{0em}
49
50 \newenvironment{labeledpar}[2]
51 {%
52     \BlockClass[%
53         \LWR@findhangingleftmargin%
54         \LWR@print@mbbox{margin-left:\LWR@printlength{\LWR@templengthone}} ; %
55         \LWR@print@mbbox{text-indent:-\LWR@printlength{\hangingindent}}%
56     ][labeledpar]#2%
57 }
58 {\endBlockClass}
59
60 \newenvironment{labeledlist}[1]
61 {\hanginglist}
62 {\endhanginglist}
63
64 \newenvironment{compactlabel}[1]
65 {\hanginglist}
66 {\endhanginglist}

```

File 123 **lwarp-hanging.sty**

§ 215 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` $\{\langle indent \rangle\} \{\langle afternum \rangle\}$

Use `hangparas` instead.

```

9 \newcommand*{\hangpara}[2]{}

```

Env `hangparas` $\{\langle indent \rangle\} \{\langle afternum \rangle\}$

```

10 \newenvironment*{hangparas}[2]
11 {%
12     \BlockClass[%
13         \LWR@print@mbx{margin-left:\LWR@printlength{#1}} ; %
14         \LWR@print@mbx{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 124 `lwarp-hypcap.sty`

§ 216 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 125 **lwarp-hypdestopt.sty**

§ 217 Package **hypdestopt**

Pkg hypdestopt **hypdestopt** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{hypdestopt}

File 126 **lwarp-hypernat.sty**

§ 218 Package **hypernat**

Pkg hypernat **hypernat** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{hypernat}

File 127 **lwarp-hyperref.sty**

§ 219 Package **hyperref**

(Emulates or patches code by SEBASTIAN RAHTZ, HEIKO OBERDIEK.)

Pkg hyperref **hyperref** is emulated.

for HTML output: 1% \LWR@ProvidesPackageDrop{hyperref}
 2 \typeout{Using the lwarp html version of package ‘hyperref’, discarding options.}
 3 \typeout{ Are not using ProvidesPackage, so that other packages}
 4 \typeout{ do not attempt to patch lwarp’s version of ‘hyperref’..}
 5% \ProvidesPackage{lwarp-#1-#2}
 6 \DeclareOption*{}
 7 \ProcessOptions\relax

 8 \newcommand*{\hypersetup}[1]{}
 9 \newcommand*{\hyperbaseurl}[1]{}

\hyperimage {\langle URL\rangle} {\langle alt text\rangle}

Insert an image with alt text:

```

10 \NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11 \LWR@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 \endgroup%
17 }
18
19 \newrobustcmd*{\hyperimage}{%
20 \begingroup%
21 \catcode'\#=12%
22 \catcode'\%=12%
23 \catcode'\&=12%
24 \catcode'\~=12%
25 \catcode'\_ =12%
26 \LWR@hyperimageb%
27 }
28

```

`\hyperdef` $\{\langle 1: category \rangle\} \{\langle 2: name \rangle\} \{\langle 3: text \rangle\}$

Creates an HTML anchor to `category.name` with the given text.

```

29 \NewDocumentCommand{\LWR@hyperdefb}{m m +m}{%
30 \LWR@ensuredoingapar%
31 \LWR@sublabel{#1.#2}%
32 #3%
33 \endgroup%
34 }
35
36 \newcommand*{\hyperdef}{%
37 \begingroup%
38 \catcode'\#=12%
39 \catcode'\%=12%
40 \catcode'\&=12%
41 \catcode'\~=12%
42 \catcode'\_ =12%
43 \LWR@hyperdefb%
44 }
45

```

`\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.

```

46 \newcommand{\LWR@hyperreffinish}[1]{%
47 \begingroup%
48 \RenewDocumentCommand{\ref}{-}{\LWR@ref@ignorestar}%

```

```

49 #1%
50 \endgroup%
51 \LWR@htmltag{/a}%
52 }
53
54 \newcommand*\LWR@hyperrefbb}[3]{%
55 \LWR@htmltag{%
56   a href="%
57     \detokenize\expandafter{#1}\LWR@hashmark%
58     \detokenize\expandafter{#2}.\detokenize\expandafter{#3}%
59   "%
60 }%
61 \endgroup%
62 \LWR@hyperreffinish%
63 }
64
65 \newrobustcmd*\LWR@hyperrefb){%
66 \begingroup%
67 \catcode'\#=12%
68 \catcode'\%=12%
69 \catcode'\&=12%
70 \catcode'\~=12%
71 \catcode'\_ =12%
72 \LWR@hyperrefbb%
73 }

```

`\LWR@hyperrefc` [*<label>*] {<text>}

Creates text as an HTML link to the \LaTeX label.

```

74
75 \NewDocumentCommand{\LWR@hyperrefcb}{0{label}}{%
76 \LWR@startref{#1}%
77 \endgroup%
78 \LWR@hyperreffinish%
79 }
80
81 \newcommand*\LWR@hyperrefc){%
82 \begingroup%
83 \catcode'\#=12%
84 \catcode'\%=12%
85 \catcode'\&=12%
86 \catcode'\~=12%
87 \catcode'\_ =12%
88 \LWR@hyperrefcb%
89 }

```

`\hyperref` {<1: URL>} {<2: category>} {<3: name>} {<4: text>} — or —

[<1: *label*>] {<2: *text*>}

```
90 \DeclareRobustCommand*\hyperref{%
91 \LWR@ensuredoingapar%
92 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
93 }
```

\hypertarget {<*name*>} {<*text*>}

Creates an anchor to *name* with the given text.

```
94 \NewDocumentCommand{\LWR@hypertargetb}{m +m}{%
95 \label{\LWR-ht-#1}%
96 #2%
97 \endgroup%
98 }
99
100 \newcommand*\hypertarget{%
101 \begingroup%
102 \catcode'\#=12%
103 \catcode'\%=12%
104 \catcode'\&=12%
105 \catcode'\~=12%
106 \catcode'\_ =12%
107 \LWR@hypertargetb%
108 }
```

\hyperlink {<*name*>} {<*text*>}

Creates a link to the anchor created by `hypertarget`, with the given link text.

Declared because also defined by **memoir**.

```
109 \DeclareDocumentCommand{\LWR@hyperlinkb}{m}{%
110 \LWR@hyperrefcb[\LWR-ht-#1]%
111 }
112
113 \DeclareDocumentCommand{\hyperlink}{l}{%
114 \LWR@ensuredoingapar%
115 \begingroup%
116 \catcode'\#=12%
117 \catcode'\%=12%
118 \catcode'\&=12%
119 \catcode'\~=12%
120 \catcode'\_ =12%
121 \LWR@hyperlinkb%
122 }
```

\autoref * {<*label*>}

For HTML, `\cleveref` is used instead.

```
123 \NewDocumentCommand{\autoref}{s m}{%
124 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}}%
125 }
```

`\autopageref` $\langle label \rangle$

For HTML, `\cleveref` is used instead.

```
126 \NewDocumentCommand{\autopageref}{s m}{%
127 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}}%
128 }
```

`\pdfstringdef` $\langle macroname \rangle$ $\langle T_{\text{E}}Xstring \rangle$

```
129 \newcommand{\pdfstringdef}[2]{}%
```

`\pdfbookmark` $[\langle level \rangle]$ $\langle text \rangle$ $\langle name \rangle$

```
130 \newcommand{\pdfbookmark}[3]{}%
```

`\currentpdfbookmark` $\langle text \rangle$ $\langle name \rangle$

```
131 \newcommand{\currentpdfbookmark}[2]{}%
```

`\subpdfbookmark` $\langle text \rangle$ $\langle name \rangle$

```
132 \newcommand{\subpdfbookmark}[2]{}%
```

`\belowpdfbookmark` $\langle text \rangle$ $\langle name \rangle$

```
133 \newcommand{\belowpdfbookmark}[2]{}%
```

`\texorpdfstring` $\langle T_{\text{E}}Xstring \rangle$ $\langle PDFstring \rangle$

```
134 \newcommand{\texorpdfstring}[2]{#1}
```

`\hypercaltcbp` $\langle dimen \rangle$ From **hyperref**.

```
135 \def\hypercaltcbp#1{%
136 \strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax
137 }%
```

`\Acrobatmenu` $\langle menuoption \rangle$ $\langle text \rangle$

```
138 \newcommand{\Acrobatmenu}[2]{}%
```

\TextField [*<parameters>*] {*<label>*}

139 \DeclareRobustCommand{\TextField}[2] [] {}

\CheckBox [*<parameters>*] {*<label>*}

140 \DeclareRobustCommand{\CheckBox}[2] [] {}

\ChoiceMenu [*<parameters>*] {*<label>*} {*<choices>*}

141 \DeclareRobustCommand{\ChoiceMenu}[3] [] {}

\PushButton [*<parameters>*] {*<label>*}

142 \DeclareRobustCommand{\PushButton}[2] [] {}

\Submit [*<parameters>*] {*<label>*}

143 \DeclareRobustCommand{\Submit}[2] [] {}

\Reset [*<parameters>*] {*<label>*}

144 \DeclareRobustCommand{\Reset}[2] [] {}

\Gauge [*<parameters>*] {*<label>*}

145 \DeclareRobustCommand{\Gauge}[2] [] {}

\LayoutTextField {*<label>*} {*<field>*}

146 \newcommand*{\LayoutTextField}[2] {}

\LayoutChoiceField {*<label>*} {*<field>*}

147 \newcommand*{\LayoutChoiceField}[2] {}

\LayoutCheckField {*<label>*} {*<field>*}

148 \newcommand*{\LayoutCheckField}[2] {}

\MakeRadioField {*<width>*} {*<height>*}

149 \newcommand*{\MakeRadioField}[2] {}

```
\MakeCheckField    {\langle width\rangle} {\langle height\rangle}
150 \newcommand*{\MakeCheckField}[2]{}
```

```
\MakeTextField    {\langle width\rangle} {\langle height\rangle}
151 \newcommand*{\MakeTextField}[2]{}
```

```
\MakeChoiceField    {\langle width\rangle} {\langle height\rangle}
152 \newcommand*{\MakeChoiceField}[2]{}
```

```
\MakeFieldButton    {\langle text\rangle}
153 \newcommand{\MakeFieldButton}[1]{}
```

File 128 **lwarp-hyperxmp.sty**

§ 220 Package **hyperxmp**

Pkg hyperxmp Emulated.

for HTML output: Discard all options for **lwarp-hyperxmp**:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 129 **lwarp-hyphenat.sty**

§ 221 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@originohyphens\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 130 **lwarp-idxlayout.sty**

§ 222 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 131 **lwarp-ifoddpagel.sty**

§ 223 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 132 **lwarp-indentfirst.sty**

§ 224 Package **indentfirst**

Pkg indentfirst **indentfirst** is ignored.

Discard all options for **lwarp-indentfirst**:

for HTML output: 1 \LWR@ProvidesPackageDrop{indentfirst}

File 133 **lwarp-inputenc.sty**

§ 225 Package **inputenc**

Pkg inputenc Error if **inputenc** is loaded after **lwarp**.

for HTML output: 1 \LWR@ProvidesPackageDrop{inputenc}

2 \LWR@loadbefore{inputenc}

File 134 **lwarp-inputenx.sty**

§ 226 Package **inputenx**

Pkg inputenx Error if **inputenx** is loaded after **lwarp**.

for HTML output: 1 \LWR@ProvidesPackageDrop{inputenx}


2 \LWR@loadbefore{inputenx}

File 135 **lwarp-keyfloat.sty**

§ 227 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}; % extra space
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 136 **lwarp-layout.sty**

§ 228 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 137 **lwarp-letterspace.sty**

§ 229 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 138 **lwarp-lettrine.sty**

§ 230 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} % extra space
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 139 **lwarp-lineno.sty**

§ 231 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=\fboxrule
104
105 \newdimen\bframesep
106 \bframesep=\fboxsep
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 140 **lwarp-lips.sty**

§ 232 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 141 **lwarp-listings.sty**

§ 233 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}

Force flexible columns:

```
3 \lst@column@flexible
```

Patches to embed listings inside pre tags:

```

4 \let\LWR@origlst@Init\lst@Init
5 \let\LWR@origlst@DeInit\lst@DeInit
6
7 \let\LWR@origlsthkEveryPar\lsthk@EveryPar
8
9 \renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{\lstlisting}{101}{#1}{#2}}
```

`\lst@Init` `{\backslash-processing}` Done at the start of a listing.

```
10 \renewcommand{\lst@Init}[1]{%
```

First, perform the **listings** initialization:

```

11 \LWR@traceinfo{\lst@Init}%
12 \renewcommand*\{@capttype}{\lstlisting}%
13 \let\lst@aboveskip\z@\let\lst@belowskip\z@%
14 \gdef\lst@boxpos{t}%
15 \let\lst@frame\@empty
16   \let\lst@frametshape\@empty
17   \let\lst@framershape\@empty
18   \let\lst@framebshape\@empty
19   \let\lst@framelshape\@empty
20 \lstframe@\lst@frameround ffff\relax%
21 \lst@multicols\@empty%
22 \LWR@origlst@Init{#1}\relax%
23 \LWR@traceinfo{finished origlst@Init}%
24 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the <pre>, then reenable line numbers.

```

25 \LWR@traceinfo{About to create verbatim.}%
26 \let\lsthk@EveryPar\relax%
27 \LWR@forcenewpage
28 \LWR@atbeginverbatim{2}{programlisting}%
29
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%
```

Inline, so open a :

```

32 \ifbool{\LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
33 \fi%
34 }
```

`\lst@DeInit` Done at the end of a listing.

```

35 \renewcommand*\{\lst@DeInit}{%
36 \lst@ifdisplaystyle%
```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenale line numbers:

```
37 \let\lsthk@EveryPar\relax%
38 \LWR@afterendverbatim{0}%
39 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
40 \else%
```

Inline, so create the closing ``:

```
41 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{</span>}}{}%
42 \fi%
```

Final **listings** deinit:

```
43 \LWR@origlst@DeInit%
44 }
```

`\lst@MakeCaption` `{</t/b>}`

This is called BOTH at the top and at the bottom of each listing.

Patched for **lwarp**.

```
45 \def\lst@MakeCaption#1{%
46 \LWR@traceinfo{MAKING CAPTION at #1}%
47 \lst@ifdisplaystyle
48 \LWR@traceinfo{making a listings display caption}%
49 \ifx #1t%
50 \ifx\lst@@caption\@empty\expandafter\lst@HRefStepCounter \else
51 \expandafter\refstepcounter
52 \fi {lstlisting}%
53 \LWR@traceinfo{About to assign label: !\lst@label!}%
54 % \ifx\lst@label\@empty\else
55 % \label{\lst@label}\fi
56 \LWR@traceinfo{Finished assigning the label.}%
57 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
58 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
59 \lst@ifnolol\else
60 \ifx\lst@@caption\@empty
61 \ifx\lst@caption\@empty
62 \ifx\lst@intname\@empty \else \def\lst@temp{ }%
63 \ifx\lst@intname\lst@temp \else
```

This code places a contents entry for a non-float. This would have to be modified for **lwarp**:

```
64 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
65 % \addcontentsline{lol}{lstlisting}{\lst@name}
66 \fi\fi
67 \fi
68 \else
```

This would have to be modified for **lwarp**:

```

69 \LWR@traceinfo{addcontents lst@caption: -\lst@caption-}%
70         \addcontentsline{lol}{lstlisting}%
71 {\protect\numberline{\thelstlisting}%
72 {\protect\ignorespaces \lst@caption \protect\relax}}%
73         \fi
74     \fi
75 \fi
76 \ifx\lst@caption\@empty\else
77 \LWR@traceinfo{lst@caption not empty-}%
78     \lst@ifsubstring #1\lst@captionpos
79         {\begingroup
80 \LWR@traceinfo{at the selected position}%

```

These space and box commands are not needed for HTML output:

```

81 %             \let\@vskip\vskip
82 %             \def\vskip{\afterassignment\lst@vskip \@tempskipa}%
83 %             \def\lst@vskip{\nobreak\@vskip\@tempskipa\nobreak}%
84 %             \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
85 %             \ifx #1t\allowbreak \fi
86             \ifx\lst@title\@empty

```

New **lwarp** code to create a caption:

```

87             \lst@makecaption\fnum\lstlisting{\ignorespaces \lst@caption}
88             \else

```

New **lwarp** code to create a title:

```

89 %             \lst@maketitle\lst@title % (AS)
90 \LWR@traceinfo{Making title: \lst@title}%
91 \begin{BlockClass}{lstlistingtitle}% lwarp
92 \lst@maketitle\lst@title% lwarp
93 \end{BlockClass}% lwarp
94     \fi
95 \LWR@traceinfo{About to assign label: !\lst@label!}%
96     \ifx\lst@label\@empty\else
97 \leavevmode% gets rid of bad space factor error
98 \GetTitleStringExpand{\lst@caption}%
99 \edef\LWR@lntemp{\GetTitleStringResult}%
100 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
101 \label{\lst@label}\fi
102 \LWR@traceinfo{Finished assigning the label.}%

```

Not needed for **lwarp**:

```

103 %             \ifx #1b\allowbreak \fi
104             \endgroup{}%
105     \fi
106 \LWR@traceinfo{end of making a listings display caption}%
107 \else
108 \LWR@traceinfo{INLINE}%

```

```

109 \fi
110 \LWR@traceinfo{DONE WITH CAPTION at #1}%
111 }

```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field \VerbatimHTMLWidth wide.

```

112 \lst@Key{numbers}{none}{%
113   \let\lst@PlaceNumber\@empty
114   \lstKV@SwitchCases{#1}%
115   {none&\\%
116     left&\def\lst@PlaceNumber{%
117 % \llap{
118 \LWR@orignormalfont%
119 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
120 % }
121 }
122 \\%
123   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
124                                     \kern\VerbatimHTMLWidth \kern\lst@numbersep
125                                     \lst@numberstyle{\thelstnumber}}}%
126   }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

127 \end{warpHTML}

```

File 142 **lwarp-longtable.sty**

§ 234 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` * [*horizontalment*] {*colspec*} Emulates the `longtable` environment.

Per the **caption** package, the starred version steps the counter per caption. The unstarred version steps the counter once at the beginning, but not at each caption.

Options [c], [l], and [r] are thrown away.

```
2 \newenvironment{longtable*}[2][{}]{%
3 \LWR@floatbegin{table}%
4 \setcaptiontype{LTcaption}%
5 \caption@setoptions{longtable}%
6 \caption@setoptions{@longtable}%
7 \caption@LT@setup%
8 \booltrue{LWR@starredlongtable}%
9 \let\captionlistentry\LWR@LTcaptionlistentry%
10 \LWR@tabular{#2}
11 }
12 {\endLWR@tabular\LWR@floatend}
13
14 \newenvironment{longtable}[2][{}]{%
15 \LWR@floatbegin{table}%
16 \setcaptiontype{LTcaption}%
17 \caption@setoptions{longtable}%
18 \caption@setoptions{@longtable}%
19 \caption@LT@setup%
20 \refstepcounter{LTcaption}%
21 \let\captionlistentry\LWR@LTcaptionlistentry%
22 \LWR@tabular{#2}
23 }
24 {\endLWR@tabular\LWR@floatend}
25
```

Provided for compatibility, but ignored:

```
26 \newcounter{LTchunksize}
27 \def\endhead{\LWR@tabularendoffline}% throws away options //[dim] and /**
28 \def\endfirsthead{\LWR@tabularendoffline}
```

```

29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand\setlongtables{}% Obsolete command, does nothing.
33 \newlength\LTleft}
34 \newlength\LTright}
35 \newlength\LTpre}
36 \newlength\LTpost}
37 \newlength\LTcapwidth}

38 \LetLtxMacro\LWR@origkill\kill
39 \renewcommand*\kill{\LWR@tabularendofline}
40 \appto\LWR@restoreorigformatting{%
41 \LetLtxMacro\kill\LWR@origkill%
42 }

```

File 143 **lwarp-lscape.sty**

§ 235 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 144 **lwarp-ltablex.sty**

§ 236 Package **ltablex**

(Emulates or patches code by ANIL K. GOEL.)

Pkg **ltablex** **ltablex** is emulated by **lwarp**.

for HTML output: Relies on **tabularx**.

```

1 \RequirePackage{tabularx}
2
3 \LWR@ProvidesPackageDrop{ltablex}
4

```

```

5 \DeclareDocumentEnvironment{tabularx}{m o m}
6 {\longtable{#3}}
7 {\endlongtable}
8
9 \DeclareDocumentEnvironment{tabularx*}{m o m}
10 {\longtable{#3}}
11 {\endlongtable}
12
13 \newcommand*{\keepXColumns}{}
14 \newcommand*{\convertXColumns}{}

```

File 145 **lwarp-ltcaption.sty**

§ 237 Package **ltcaption**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg ltcaption **ltcaption** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltcaption}

\LTcapttype is already defined by **lwarp**.

longtable* is already defined by **lwarp-longtable**.

```

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}
5 \newcommand*{\LTcapmarginfalse}{}

```

File 146 **lwarp-ltxgrid.sty**

§ 238 Package **ltxgrid**

Pkg ltxgrid **ltxgrid** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltxgrid}

```


2 \newcommand*{\onecolumngrid}{}
3 \newcommand*{\twocolumngrid}{}
4 \newcommand*{\removestuff}{}
5 \newcommand*{\addstuff}[2]{}
6 \newcommand*{\replacestuff}[2]{}

```

File 147 **lwarp-ltxtable.sty**

§ 239 Package **ltxtable**

Pkg ltxtable **ltxtable** is emulated.

 **table numbering** The print version does not seem to honor longtable* from the **caption** package, while **lwarp** does.

for HTML output: 1 \LWR@ProvidesPackageDrop{ltxtable}

```
\LTXtable    {\width}\{file}

2 \newcommand*\LTXtable[2]{%
3 \input{#2}%
4 }
```

File 148 **lwarp-lua-check-hyphen.sty**

§ 240 Package **lua-check-hyphen**

Pkg lua-check-hyphen **lua-check-hyphen** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{lua-check-hyphen}

```
2 \newcommand*\LuaCheckHyphen[1]{}
```

File 149 **lwarp-luacolor.sty**

§ 241 Package **luacolor**

Pkg luacolor **luacolor** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{luacolor}

```
2 \newcommand{\luacolorProcessBox}[1]{}
```

File 150 **lwarp-luatodonotes.sty**

§ 242 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}{l@do}{#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:\LWR@origpound{}B3FFB3]{highlight}{#1}%
82 }
83
84 \fi% \if@todonotes@disabled

```

File 151 **lwarp-magaz.sty**

§ 243 Package **magaz**

Pkg magaz **magaz** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{magaz}

2 \newcommand\FirstLine[1]{%
3   \begingroup%
4   \FirstLineFont{%
5     \LWR@textcurrentcolor{%
6       \LWR@textcurrentfont{%
7         #1%
8       }%
9     }%
10  }%
11  \endgroup%
12 }
13
14 \providecommand\FirstLineFont{\scshape}

```

File 152 **lwarp-marginfit.sty**

§ 244 Package **marginfit**

Pkg marginfit **marginfit** is ignored.

for HTML output: Discard all options for **lwarp-marginfit**:

```

1 \LWR@ProvidesPackageDrop{marginfit}

```

File 153 **lwarp-marginfix.sty**

§ 245 Package **marginfix**

(Emulates or patches code by STEPHEN HICKS.)

Pkg marginfix Emulated.

for HTML output: Discard all options for **lwarp-marginfix**:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][ ]{}
10 \newcommand*{\unblockmargin}[1][ ]{}
11 \newcommand*{\marginphantom}[2][ ]{}

```

File 154 **lwarp-marginnote.sty**

§ 246 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 155 **lwarp-mcaption.sty**

§ 247 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 156 **lwarp-mdframed.sty**

§ 248 Package **mdframed**

(Emulates or patches code by MARCO DANIEL, ELKE SCHUBERT.)

Pkg mdframed **mdframed** is loaded with options forced to framemethod=none.

§ 248.1 **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.

§ 248.2 Package loading

for HTML output:

```
1 \RequirePackage{xcolor}% for \convertcolorspec
2
3 \LWR@ProvidesPackageDrop{mdframed}
```

amsthm must be loaded before **mdframed**

```
4 \LWR@origRequirePackage{amsthm}
```

Do not require Tikz or pstricks:

```
5 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

§ 248.3 Patches

Patch to remove PDF formatting and add HTML tags:

```
6 \AtBeginDocument{
7 \def\mdf@trivlist#1{%
8   \edef\mdf@temp{%
9     \topsep=\the\topsep\relax%
10    \partopsep=\the\partopsep\relax%
11    \parsep=\the\parsep\relax%
12  }%
13    \setlength{\topsep}{#1}%
14    \topskip\z@%
15    \partopsep\z@%
16    \parsep\z@%
```

```

17 % \@nmbrlistfalse%
18 % \@trivlist%
19 % \labelwidth\z@%
20 % \leftmargin\z@%
21 % \itemindent\z@%
22 \let\@itemlabel\@empty%
23 \def\makelabel##1{##1}%
24 % \item\relax\mdf@temp\relax%
25 }
26
27 \renewcommand*{\endmdf@trivlist}{%
28 \LWR@traceinfo{endmdf@trivlist}%
29 % \endtrivlist%
30 \LWR@listend%
31 }
32 }% AtBeginDocument

```

§ 248.4 Initial setup

To handle CSS and paragraphs, patch code at start and end of environment and contents. \LWR@origraggedright helps avoid hyphenation.

```

33 \mdfsetup{
34 startcode={\LWR@mdframedstart\LWR@origraggedright},
35 endcode={\LWR@mdframedend},
36 startinnercode={\LWR@startpars\LWR@origraggedright},
37 endinnercode={\LWR@stoppars},
38 }

```

§ 248.5 Color and length HTML conversion

\LWR@mdfprintcolor *{\langle mdfcolorkey \rangle}*

Given the **mdframed** key, print the color.

```

39 \newcommand*{\LWR@mdfprintcolor}[1]{%
40 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
41 \LWR@origpound\LWR@tempcolor
42 }

```

\LWR@mdfprintlength *{\langle mdflengthkey \rangle}*

Given the **mdframed** key, print the length.

```

43 \newcommand*{\LWR@mdfprintlength}[1]{%
44 \LWR@printlength{\csuse{mdf@#1@length}}
45 }

```

§ 248.6 Environment encapsulation

`\LWR@mdframedstart` Actions before an mdframe starts.

Encapsulate a frame inside a `<div>` of the desired class.

```
46 \newcommand*{\LWR@mdframedstart}{%
47 \LWR@traceinfo{\LWR@mdframedstart start}%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
48 \LWR@stoppars%
```

Open a `<div>` and with custom class and custom style:

```
49 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
50 style=" \LWR@orignewline
```

Convert and print the background color:

```
51 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
52 border: \LWR@mdfprintlength{linewidth} solid
53 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
54 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
55 \ifbool{mdf@shadow}{%
56   box-shadow:
57     \LWR@mdfprintlength{shadowsize}
58     \LWR@mdfprintlength{shadowsize}
59     \LWR@mdfprintlength{shadowsize}
60     \LWR@mdfprintcolor{shadowcolor} ;
61 }
62 {box-shadow: none ;}
63 \LWR@orignewline
```

```
64 "}
```

```
65 % \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`:

```
66 \LWR@select@print@hspace%
67 \renewcommand*{\rule}{\LWR@print@rule}
68 \LetLtxMacro\makebox\LWR@print@makebox%

69 \LWR@startpars%
70 \LWR@traceinfo{\LWR@mdframedstart done}%
71 }
```

`\LWR@mdframedend` Actions after an mdframe ends.

After closing the `<div>`, globally restore to the default environment type:

```
72 \newcommand*\LWR@mdframedend}{
73 \LWR@traceinfo{\LWR@mdframedend start}%
```

Close the custom `<div>`:

```
74 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
75 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
76 \LWR@startpars%
77 \LWR@traceinfo{\LWR@mdframedend done}%
78 }
```

§ 248.7 Mdfamed environment

```
79 \renewenvironment{mdframed}[1][\fi]{%
80 \color@begingroup%
81 \mdfsetup{userdefinedwidth=\linewidth,#1}%
82 \mdf@startcode%
83 \mdf@preenvsetting%
84 \ifdefempty{\mdf@firstframetitle}{\fi}%
85 \let\mdf@frametitlesave\mdf@frametitle%
86 \let\mdf@frametitle\mdf@firstframetitle%
87 }%
88 \ifvmode\nointerlineskip\fi%
89 \ifdefempty{\mdf@frametitle}{\fi}%
90 {\mdfframedtitleenv{\mdf@frametitle}%
91 \mdf@frametitle@use%
92 }%
93 \mdf@trivlist{\mdf@skipabove@length}%%
94 \mdf@settings%
95 \mdf@lrbox{\mdf@splitbox@one}%
96 \mdf@startinnercode%
97 }%
98 {%
99 \mdf@ignorelastdescenders%
100 \par%
101 \unskip\ifvmode\nointerlineskip\hrule \@height\z@ \@width\hsize\fi%
102 \ifmdf@footnoteinside%
103 \def\mdf@reserveda{%
104 \mdf@footnoteoutput%
105 \mdf@endinnercode%
106 \endmdf@lrbox%
107 \ifdefempty{\mdf@frametitle}{\fi}%
```

```

108 %           {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
109 %           \detected@mdf@put@frame
110 %       }%
111 %   \else%
112 %       \def\mdf@reserveda{%
113 %           \mdf@endinnercode%
114 %           \endmdf@lrbox%
115 %           \ifdefempty{\mdf@frametitle}{}%
116 %           {\mdfframedtitleenv{\mdf@frametitle}\mdf@@frametitle@use}%
117 %           \detected@mdf@put@frame%
118 %           \mdf@footnoteoutput%
119 %       }%
120 %   \fi%
121 %   \mdf@reserveda%
122 %   \aftergroup\endmdf@trivlist%
123 %   \color@endgroup%
124 %   \mdf@endcode%
125 }

```

\mdf@footnoteoutput

```

126 \renewrobustcmd*\mdf@footnoteoutput{%
127     \LWR@printpendingmpfootnotes%
128 }

```

§ 248.8 Titles and subtitles

\mdfframedtitleenv {<title>}

Encapsulation of the original which places the title inside a of class mdfframedtitle:

```

129 \LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv
130
131 \newlength{\LWR@titleroundcorner}
132
133 \renewrobustcmd\mdfframedtitleenv[1]{%
134 \LWR@traceinfo{\LWR@mdfframedtitleenv start}%
135 % \LWR@origmdfframedtitleenv{%

```

Open a with a custom class and custom style:

```

136 \LWR@htmltagc{span class="mdfframedtitle" \LWR@orignewline
137 style=" \LWR@orignewline

```

Convert and print the title background color:

```

138 background:
139 \LWR@mdfprintcolor{frametitlebackgroundcolor}
140 ; \LWR@orignewline

```

Convert and print the title rule:

```

141 \ifbool{mdf@frametitlerule}{%

```

```

142 border-bottom:
143   \LWR@mdfprintlength{frametitlerulewidth}
144   solid
145   \LWR@mdfprintcolor{frametitlerulecolor}
146   ; \LWR@orignewline
147 }{}%

```

Finish the custom style and the opening span tag:

```

148 " \LWR@orignewline
149 }% span

```

Restrict paragraph tags inside a span:

```

150 \begin{LWR@nestspan}%

```

Print the title inside the span:

```

151 #1%

```

Close the span and unnest the paragraph tag restriction:

```

152 \LWR@htmltagc{/span}%
153 \end{LWR@nestspan}%
154 % }
155 \LWR@traceinfo{LWR@mdframedtitleenv end}%
156 }

```

```

\LWR@mdfsubtitlecommon {\sub -or- subsub} [\langle options \rangle] {\langle title \rangle}

```

Common code for \LWR@mdfsubtitle and \LWR@mdfsubsubtitle.

Encapsulate the subtitle inside a of class mdframedsubtitle:

```

157 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
158 {% the following empty line is required
159
160 \LWR@traceinfo{LWR@mdframedsubtitlecommon start}%

```

Special handling for mdframed: Subtitles have \pars around them, so temporarily disable them here.

```

161 \let\par\LWR@origpar%

```

Open a with a custom class and custom style:

```

162 \LWR@htmltagc{span class="mdframed#1title"
163 style=" \LWR@orignewline

```

Convert and print the background color:

```

164 background:
165 \LWR@mdfprintcolor{#1titlebackgroundcolor}
166 ; \LWR@orignewline

```

Convert and print the above line:

```

167 \ifbool{mdf@#1titleaboveline}{%
168   border-top:
169   \LWR@mdfprintlength{#1titleabovelinewidth}
170   solid
171   \LWR@mdfprintcolor{#1titleabovelinecolor}
172   ; \LWR@orignewline
173 }{}%

```

Convert and print the below line:

```

174 \ifbool{mdf@#1titlebelowline}{%
175   border-bottom:
176   \LWR@mdfprintlength{#1titlebelowlinewidth}
177   solid
178   \LWR@mdfprintcolor{#1titlebelowlinecolor}
179   ; \LWR@orignewline
180 }{}%

```

Finish the custom style and the opening span tag:

```

181 "% span

```

Restrict paragraph tags inside a span:

```

182 \begin{LWR@nestspan}%

```

Perform the original subtitle action:

```

183 \IfNoValueTF{#2}
184 {\csuse{LWR@origmdf#1title}{#3}}%
185 {\csuse{LWR@origmdf#1title}[#2]{#3}}%

```

Close the span and unnest the paragraph tag restriction:

```

186 \LWR@htmltagc{/span}% the following empty line is required
187 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
188
189 \LWR@traceinfo{LWR@mdframedsubtitlecommon end}%
190 }

```

`\LWR@mdfsubtitle` [*<options>*] {*<title>*}

```

191 \newcommand*{\LWR@mdfsubtitle}{%
192 \LWR@mdfsubtitlecommon{sub}%
193 }
194 \let\mdfsubtitle\LWR@mdfsubtitle

```

`\LWR@mdfsubsubtitle` [*<options>*] {*<title>*}

```

195 \newcommand*{\LWR@mdfsubsubtitle}{%
196 \LWR@mdfsubsubtitlecommon{subsub}%
197 }
198 \let\mdfsubsubtitle\LWR@mdfsubsubtitle

```

§ 248.9 New environments

`\LWR@mdthisenv` Stores the environment of the frame about to be created:

```
199 \newcommand*{\LWR@mdthisenv}{mdframed}
```

`\newmdenv` [*<options>*] {*<env-name>*}

Modified from the original to remember the environment.

```
200 \renewrobustcmd*{\newmdenv}[2][{}]{%
201 \newenvironment{#2}{%
202 {%
203 \mdfsetup{#1}%
204 \renewcommand*{\LWR@mdthisenv}{md#2}%
205 \begin{mdframed}%
206 }
207 {\end{mdframed}}}%
208 }
```

`\surroundwithmdframed` [*<options>*] {*<environment>*}

Modified from the original to remember the environment.

```
209 \renewrobustcmd*{\surroundwithmdframed}[2][{}]{%
210 \BeforeBeginEnvironment{#2}{%
211 \renewcommand*{\LWR@mdthisenv}{md#2}%
212 \begin{mdframed}[#1]}%
213 \AfterEndEnvironment{#2}{\end{mdframed}}}%
214 }
```

`\mdtheorem` [*<mdframed-options>*] {*<envname>*} [*<numberedlike>*] {*<caption>*} [*<within>*]

Modified from the original to remember the environment.

```
215 \DeclareDocumentCommand{\mdtheorem}{0}{m o m o }{%
216 {\ifcsdef{#2}%
217 {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
218 {%
219 \IfNoValueTF {#3}%
220 {%#3 not given -- number relationship
221 \IfNoValueTF {#5}%
222 {%#3+#5 not given
223 \@definecounter{#2}%
224 \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
225 \newenvironment{#2}[1][{}]{%
226 \refstepcounter{#2}%
227 \ifstrempy{##1}%
228 {\let\@temptitle\relax}%
229 {%
230 \def\@temptitle{\mdf@theoremseparator%
231 \mdf@theoremspace%
```

```

232             \mdf@theoremtitlefont%
233             ##1}%
234         \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
235     }%
236     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
237                                     \@temptitle}}}%
238     {\end{mdframed}}%
239     \newenvironment{#2*}[1][]{%
240         \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
241         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}}}%
242         {\end{mdframed}}%
243     }%
244     {%#5 given -- reset counter
245     \@definecounter{#2}\@newctr{#2}[#5]%
246     \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
247     \expandafter\xdef\csname the#2\endcsname{%
248         \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
249         \@thmcounter{#2}}%
250     \newenvironment{#2}[1][]{%
251         \refstepcounter{#2}%
252         \ifstrempy{##1}%
253             {\let\@temptitle\relax}%
254             {%
255                 \def\@temptitle{\mdf@theoremseparator%
256                                 \mdf@theoremspace%
257                                 \mdf@theoremtitlefont%
258                                 ##1}%
259                 \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
260             }
261         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
262                                     \@temptitle}}}%
263         {\end{mdframed}}%
264     \newenvironment{#2*}[1][]{%
265         \ifstrempy{##1}%
266             {\let\@temptitle\relax}%
267             {%
268                 \def\@temptitle{\mdf@theoremseparator%
269                                 \mdf@theoremspace%
270                                 \mdf@theoremtitlefont%
271                                 ##1}%
272                 \mdf@thm@caption{#2}{#{#4}{\csname the#2\endcsname}{##1}}%
273             }%
274         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}}}%
275         {\end{mdframed}}%
276     }%
277 }%
278 {%#3 given -- number relationship
279     \global\@namedef{the#2}{\@nameuse{the#3}}%
280     \newenvironment{#2}[1][]{%
281         \refstepcounter{#3}%

```

```

282     \ifstrempy{##1}%
283     {\let\@temptitle\relax}%
284     {%
285         \def\@temptitle{\mdf@theoremseparator%
286                         \mdf@theoremspace%
287                         \mdf@theoremtitlefont%
288                         ##1}%
289         \mdf@thm@caption{#2}{\csname the#2\endcsname}{##1}}%
290     }
291     \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
292                                     \@temptitle}]]%
293     {\end{mdframed}}%
294     \newenvironment{#2*}[1][]{%
295         \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{\ \ ##1}}%
296         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
297         {\end{mdframed}}%
298     }%
299     \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}% lwarp
300     \BeforeBeginEnvironment{#2*}{\renewcommand*\LWR@mdthisenv}{md#2}}% lwarp
301 }%
302 }

```

`\newmdtheoremenv` [*⟨mdframed-options⟩*] {*⟨envname⟩*} [*⟨numberedlike⟩*] {*⟨caption⟩*} [*⟨within⟩*]

Modified from the original to remember the environment.

```

303 \DeclareDocumentCommand\newmdtheoremenv{0}{m o m o }{%
304   \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }{%
305     {\newtheorem{#2}{#4}}%
306     {%
307       \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
308       \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
309     }%
310   \BeforeBeginEnvironment{#2}{%
311     \renewcommand*\LWR@mdthisenv{md#2}%
312     \begin{mdframed}[#1]}%
313   \AfterEndEnvironment{#2}{%
314     \end{mdframed}}%
315 }

```

File 157 **lwarp-memhfixc.sty**

§ 249 Package **memhfixc**

 Pkg `memhfixc` **memhfixc** is ignored.

 for HTML output: 1 \LWR@ProvidesPackageDrop{memhfixc}

File 158 **lwarp-metalogo.sty**

§ 250 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 159 **lwarp-mhchem.sty**

§ 251 Package **mhchem**

(Emulates or patches code by MARTIN HENSEL.)

Pkg mhchem **mhchem** is patched for use by **lwarp**.


mhchem expressions are converted to svg math. Inline expressions use hashed filenames to allow reuse, and assume that any **mhchem** options are global.

 **MATHJAX and mhchem** The MATHJAX **mhchem** extension is not yet used. If MATHJAX is used for math in the rest of the document, **lwarp** converts standalone **mhchem** expressions into svg math images, but expressions inside math must be placed between `\displaymathother` and `\displaymathnormal`:

```

\displaymathother
\[ \ce{ ... } \] ... $ \ce { ... } $
\displaymathnormal

```

 **nested math** When producing HTML output, **lwarp** does not support the use of nested dollar signs in **mhchem** expressions.

For some examples from the **mhchem** manual, change as follows:

$\text{\ce{NaOH(aq, ∞)}}$	% old
$\text{\ce{NaOH(aq, ∞)}}$	% new
$\text{\ce{Fe(CN)_{\frac{6}{2}}}}$	% old
$\text{\ce{Fe(CN)_{\frac{6}{2}}}}$	% new
$\text{\ce{NO_x}}$	% old
$\text{\ce{NO_x}}$	% new
$\text{\ce{NO_x}}$	% old
$\text{\ce{NO_x}}$	% new
$\text{\ce{\textit{cis} [PtCl2(NH3)2]}}$	% old
$\text{\ce{\textit{cis} [PtCl2(NH3)2]}}$	% new

for HTML output: `1 \LWR@ProvidesPackagePass{mhchem}`

The original definition of `\ce`:

```
2 \LetLtxMacro\LWR@mhchem@origce\ce
```

The new definition, called from the new `\ce` after math shift is set. The starred `lateximage` uses a hashed filename for the svg image. The alt tag is set to the **mhchem** expression.

```
3 \newcommand{\LWR@mhchem@HTML@ce}[1]{%
4 \begin{lateximage}*[ \textbackslash{}ce{\LWR@HTMLsanitize{#1}}]%
5 \LWR@mhchem@origce{#1}%
6 \end{lateximage}%
7 \endgroup%
8 \addtocounter{LWR@mhchem@cedepth}{-1}%
9 }
```

Only set math shift if outer depth:

```
10 \newcounter{LWR@mhchem@cedepth}
11 \setcounter{LWR@mhchem@cedepth}{0}
```

The new `\ce`. Sets math shift then continues.

```
12 \renewcommand{\ce}{%
13 \begingroup%
14 \ifnumequal{\value{LWR@mhchem@cedepth}}{0}{%
15   \catcode'\$=3% math shift
16 }{}%
17 \addtocounter{LWR@mhchem@cedepth}{1}%
18 \LWR@mhchem@HTML@ce%
```

```
19 }
```

The original definition of `\cesplit`:

```
20 \LetLtxMacro\LWR@mhchem@origcesplit\cesplit
```

The new definition, called from the new `\cesplit` after math shift is set. The starred `lateximage` uses a hashed filename for the svg image. The alt tag is set to the **mhchem** expression.

```
21 \newcommand*{\LWR@mhchem@HTML@cesplit}[2]
22 {%
23 \begin{lateximage}*[\textbackslash{}cesplit\{\LWR@HTMLsanitize{#2}\}]%
24 \LWR@mhchem@origcesplit{#1}{#2}%
25 \end{lateximage}%
26 \endgroup%
27 }
```

Only set math shift if outer depth:

```
28 \newcounter{LWR@mhchem@cesplitdepth}
29 \setcounter{LWR@mhchem@cesplitdepth}{0}
```

The new `\cesplit`. Sets math shift then continues.

```
30 \renewcommand{\cesplit}{%
31 \begingroup%
32 \ifnumequal{\value{LWR@mhchem@cesplitdepth}}{0}{%
33   \catcode'\$=3% math shift
34 }{}%
35 \addtocounter{LWR@mhchem@cesplitdepth}{1}%
36 \LWR@mhchem@HTML@cesplit%
37 }
```

Resore originals inside a `lateximage`:

```
38 \appto\LWR@restoreorigformatting{%
39 \LetLtxMacro\ce\LWR@mhchem@origce%
40 \LetLtxMacro\cesplit\LWR@mhchem@origcesplit%
41 }
```

File 160 **lwarp-microtype.sty**

§ 252 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 161 **lwarp-midfloat.sty**

§ 253 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 162 **lwarp-midpage.sty**

§ 254 Package **midpage**

Pkg midpage **midpage** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{midpage}

```
2 \newenvironment{midpage}
3 {\begin{BlockClass}[\LWR@print@mbbox{margin-top:6ex} ; \LWR@print@mbbox{margin-bottom:6ex}]{midpa
4 {\end{BlockClass}}}
```

File 163 **lwarp-morefloats.sty**

§ 255 Package **morefloats**

Pkg morefloats **morefloats** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{morefloats}

File 164 **lwarp-moreverb.sty**

§ 256 Package **moreverb**

(Emulates or patches code by ROBIN FAIRBAIRNS.)

Pkg moreverb **moreverb** is supported with some patches.

for HTML output: 1 \begin{warpHTML}

```
2 \LWR@ProvidesPackagePass{moreverb}

3 \BeforeBeginEnvironment{verbatimtab}{%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{3}{Verbatim}%
6 }
7 \AfterEndEnvironment{verbatimtab}{%
```

```

8 \LWR@afterendverbatim{1}%
9 }
10
11
12 \LetLtxMacro\LWRMV@orig@verbatimtabinput\@verbatimtabinput
13
14 \renewcommand{\@verbatimtabinput}[2] [] {%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{3}{Verbatim}%
17 \LWRMV@orig@verbatimtabinput[#1]{#2}%
18 \LWR@afterendverbatim{1}%
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
23 \LWR@atbeginverbatim{3}{programlisting}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 \LWR@afterendverbatim{1}%
28 }
29
30 \BeforeBeginEnvironment{listingcont}{%
31 \LWR@forcenewpage
32 \LWR@atbeginverbatim{3}{programlisting}%
33 }
34
35 \AfterEndEnvironment{listingcont}{%
36 \LWR@afterendverbatim{1}%
37 }

38 \LetLtxMacro\LWRMV@@listinginput\@listinginput
39
40 \renewcommand{\@listinginput}[3] [] {
41 \LWR@forcenewpage
42 \LWR@atbeginverbatim{3}{programlisting}%
43 \LWRMV@@listinginput[#1]{#2}{#3}%
44 \LWR@afterendverbatim{1}%
45 }
46
47
48 \renewenvironment*{boxedverbatim}
49 {
50 \LWR@forcenewpage
51 \LWR@atbeginverbatim{3}{boxedverbatim}%
52 \verbatim%
53 }
54 {
55 \endverbatim%

```

```
56 \LWR@afterendverbatim{1}%
57 }
```

```
58 \end{warpHTML}
```

File 165 **lwarp-morewrites.sty**

§ 257 Package **morewrites**

Pkg **morewrites** Error if **morewrites** is loaded after **lwarp**.

Discard all options for **lwarp-morewrites**:

for HTML output: 1 \LWR@ProvidesPackageDrop{morewrites}

 2 \LWR@loadbefore{morewrites}

File 166 **lwarp-mparhack.sty**

§ 258 Package **mparhack**

Pkg **mparhack** Ignored.

for HTML output: Discard all options for **lwarp-mparhack**:

 1 \LWR@ProvidesPackageDrop{mparhack}

File 167 **lwarp-multicol.sty**

§ 259 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 * {\langle numcols \rangle} [\langle heading \rangle]
3 \NewDocumentEnvironment{multicols}{s m o}
```

HTML `<div>` class to contain everything:

```
4 {
5 \LWR@forcenewpage
6 \BlockClass{multicols}
```

Optional HTML `<div>` class for the heading:

```
7 \IfValueT{#3}{\begin{BlockClass}{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 168 **lwarp-multirow.sty**

§ 260 Package **multirow**

(Emulates or patches code by PIET VAN OOSTRUM, ØYSTEIN BACHE, JERRY LEICHTER.)

Pkg `multirow` **multirow** is emulated during HTML output, and used as-is while inside a `lateximage`.

In a `lateximage`, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 68.23 for the print-mode versions.

for HTML output: Remove the placeholder macro which was used if **multirow** was not loaded:

```
1 \LetLtxMacro\multirow\relax
2 \LWR@ProvidesPackagePass{multirow}
```

`\LWR@multirowborder` Set to left or right to create a thick border for the cell, for use by **bigdelim**:

```
3 \newcommand{\LWR@multirowborder}{}%
```

§ 260.1 Multirow

`\multirow` [*⟨vpos⟩*] {*⟨numrows⟩*} [*⟨bigstruts⟩*] {*⟨width⟩*} [*⟨fixup⟩*] {*⟨text⟩*}

```
4 \NewDocumentCommand{\LWR@HTML@multirow}{O{c} m o m o +m}%
5 {%
6 \LWR@traceinfo{*** LWR@HTML@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@print@mbbox{vertical-align:bottom}" }{}%
12 \ifstrequal{#1}{t}{style="\LWR@print@mbbox{vertical-align:top}" }{}%
13 }%
```

The left/right border, if given:

```
14 \ifdefvoid{\LWR@multirowborder}{}{%
15 style="\LWR@print@mbbox{border-\LWR@multirowborder:} 2px dotted black ; %
16 \LWR@print@mbbox{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@addcdashline%
27 \LWR@addtabularrulecolors%
28 \LWR@tdendstyles%
29 }%
```

The column's < spec:

```
30 \LWR@getexparray{\LWR@colbefore}{{\arabic{\LWR@tablecolindex}}}%

```

While printing the text, redefine `\` to generate a new line

```
31 \begingroup\LetLtxMacro{\}\{\LWR@endoffline}\#6\endgroup%
32 \LWR@stoppars%
33 \global\boolfalse{\LWR@intabularmetadata}%
34 \renewcommand{\LWR@multirowborder}{}%
35 \LWR@traceinfo{*** LWR@HTML@multirow done}%
36 }%
37
38 \LWR@formatted{multirow}
```

§ 260.2 Combined multicolumn and multirow

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

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

The two arguments for `\multicolumn` come first, followed by the five arguments for `\multirow`, many of which are optional, followed by the contents.

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

⚠ **empty cells**

```

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

```

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

```

\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}

```

`\@ifpackageloaded{multirow}` determines if v2.0 or later of **multirow** was used, which included the `\ProvidesPackage` macro.

The HTML version follows.

`\AtBeginDocument` because the print version had to see if **multirow** was loaded before determining how to define `\LWR@print@multicolumnrow`.

```

39 \AtBeginDocument{
40
41 \NewExpandableDocumentCommand{\LWR@HTML@multicolumnrow}{m m O{} m O{} m O{} +m}{%

```

Figure out how many extra HTML columns to add for @ and ! columns:

```

42 \LWR@tabularhtmlcolumns{\arabic{LWR@tablecolindex}}{#1}

```

Create the multicolumn/multirow tag:

```

43 \begingroup%
44 \LetLtxMacro{\}\{\LWR@endofline}%
45 \LWR@domulticolumn[#3][#4]{#1}{\arabic{LWR@tabhtmlcoltotal}}{#2}{#8}%
46 \endgroup%

```

Move to the next \LaTeX column:

```

47 \addtocounter{LWR@tablecolindex}{#1}%
48 \addtocounter{LWR@tablecolindex}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```

49 \booltrue{LWR@skipatbang}%
50 }
51
52 \LWR@expandableformatted{multicolumnrow}
53
54 }% \AtBeginDocument

```

File 169 **lwarp-multitoc.sty**

§ 261 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 170 **lwarp-nameref.sty**

§ 262 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 171 **lwarp-natbib.sty**

§ 263 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 172 **lwarp-needspace.sty**

§ 264 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 173 **lwarp-newclude.sty**

§ 265 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 174 **lwarp-newunicodechar.sty**

§ 266 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 175 **lwarp-nextpage.sty**

§ 267 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}{}

File 176 **lwarp-nicefrac.sty**

§ 268 Package **nicefrac**

(Emulates or patches code by AXEL REICHERT.)

Pkg nicefrac **nicefrac** is patched for use by **lwarp**.

for HTML output: 1 \LWR@ProvidesPackagePass{nicefrac}[1998/08/04]

nicefrac uses \TeX boxes, so `\@ensuredmath` must be restored temporarily:

```
2 \LetLtxMacro\LWR@origUnitsNiceFrac\@UnitsNiceFrac
3
4 \DeclareRobustCommand*\@UnitsNiceFrac[3] [] {%
5 \begingroup%
6 \LetLtxMacro\@ensuredmath\LWR@origensuredmath%
7 \LWR@origUnitsNiceFrac[#1]{#2}{#3}%
8 \endgroup%
9 }
```

File 177 **lwarp-nonfloat.sty**

§ 269 Package **nonfloat**

(Emulates or patches code by KAI RASCHER.)

Pkg nonfloat **nonfloat** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonfloat}

```
2 \LetLtxMacro\topcaption\caption
3 \newcommand{\figcaption}{\def\@capttype{figure}\caption}
4 \newcommand{\tabcaption}{\def\@capttype{table}\topcaption}
5 \newenvironment{narrow}[2]{}{}
```

File 178 **lwarp-nonumonpart.sty**

§ 270 Package **nonumonpart**

Pkg nonumonpart **nonumonpart** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

File 179 **lwarp-nopageno.sty**

§ 271 Package **nopageno**

Pkg nopageno **nopageno** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

File 180 **lwarp-nowidow.sty**

§ 272 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 181 **lwarp-nttheorem.sty**

§ 273 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.

§ 273.1 **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.

§ 273.2 **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 \LWR@ProvidesPackagePass{ntheorem}

```

§ 273.3 Remembering the theorem style

Storage for the style being used for new theorems.

```

23 \newcommand{\LWR@newtheoremstyle}{plain}

24 \AtBeginDocument{
25 \ifpackageloaded{cleveref}{
26 \gdef\@thm#1#2#3{%
27   \if@thmmarks
28     \stepcounter{end\InTheoType ctr}%
29   \fi
30   \renewcommand{\InTheoType}{#1}%
31   \if@thmmarks
32     \stepcounter{curr#1ctr}%
33     \setcounter{end#1ctr}{0}%
34   \fi
35   \refstepcounter[#1]{#2}% <<< cleveref modification
36   \theorem@prework
37   \LWR@forcenewpage% lwarp
38   \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
39   \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
40   \ifuse@newframeskips % cf. latex.ltx for topsepadd: \@trivlist
41     \ifthm@inframe
42       \thm@topsep\theoreminframepreskipamount
43       \thm@topsepadd\theoreminframepostskipamount
44     \else
45       \thm@topsep\theorempreskipamount
46       \thm@topsepadd\theorempostskipamount

```

```

47     \fi
48     \else% oldframeskips
49         \thm@topsep\theorempreskipamount
50         \thm@topsepadd \theorempostskipamount
51         \ifvmode\advance\thm@topsepadd\partopsep\fi
52     \fi
53     \@topsep\thm@topsep
54     \@topsepadd\thm@topsepadd
55     \advance\linewidth -\theorem@indent
56     \advance\linewidth -\theorem@rightindent
57     \advance\@totalleftmargin \theorem@indent
58     \parshape \@ne \@totalleftmargin \linewidth
59     \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
60 }
61 }{% not @ifpackageloaded{cleveref}
62 \gdef\@thm#1#2#3{%
63     \if@thmmarks
64         \stepcounter{end\InTheoType ctr}%
65     \fi
66     \renewcommand{\InTheoType}{#1}%
67     \if@thmmarks
68         \stepcounter{curr#1ctr}%
69         \setcounter{end#1ctr}{0}%
70     \fi
71     \refstepcounter{#2}%
72     \theorem@prework
73     \LWR@forcenewpage% lwarp
74     \BlockClass{theorembody#1}%\LWR@thisthmstyle% lwarp
75     \trivlist % latex's \trivlist, calling latex's \@trivlist unchanged
76     \ifuse@newframeskips % cf. latex.ltx for topsepadd: \@trivlist
77         \ifthm@inframe
78             \thm@topsep\theoreminframepreskipamount
79             \thm@topsepadd\theoreminframepostskipamount
80         \else
81             \thm@topsep\theorempreskipamount
82             \thm@topsepadd\theorempostskipamount
83         \fi
84     \else% oldframeskips
85         \thm@topsep\theorempreskipamount
86         \thm@topsepadd \theorempostskipamount
87         \ifvmode\advance\thm@topsepadd\partopsep\fi
88     \fi
89     \@topsep\thm@topsep
90     \@topsepadd\thm@topsepadd
91     \advance\linewidth -\theorem@indent
92     \advance\linewidth -\theorem@rightindent
93     \advance\@totalleftmargin \theorem@indent
94     \parshape \@ne \@totalleftmargin \linewidth
95     \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}
96 }

```

```

97 }
98 }% AtBeginDocument

```

Patched to remember the style being used for new theorems:

```

99 \gdef\theoremstyle#1{%
100   \@ifundefined{th@#1}{\@warning
101     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
102     \theoremstyle{plain}
103     \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
104   }%
105   {
106     \theoremstyle{#1}
107     \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
108   }
109 }

```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```

110
111 \gdef\xnthm#1#2[#3]{%
112   \ifthm@tempif
113     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
114     \expandafter\@ifundefined{c@#1}%
115       {\@definecounter{#1}}{}%
116     \@newctr{#1}[#3]%
117     \expandafter\xdef\csname the#1\endcsname{%
118       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
119       {\noexpand\csname the\theoremnumbering\endcsname{#1}}}%
120     \expandafter\gdef\csname mkheader@#1\endcsname
121       {\csname setparams@#1\endcsname
122        \@thm{#1}{#1}{#2}
123       }%
124     \global\@namedef{end#1}{\@endtheorem}
125     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
126   \fi
127 }
128
129 \gdef\ynthm#1#2{%
130   \ifthm@tempif
131     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
132     \expandafter\@ifundefined{c@#1}%
133       {\@definecounter{#1}}{}%
134     \expandafter\xdef\csname the#1\endcsname
135       {\noexpand\csname the\theoremnumbering\endcsname{#1}}%
136     \expandafter\gdef\csname mkheader@#1\endcsname
137       {\csname setparams@#1\endcsname
138        \@thm{#1}{#1}{#2}

```

```

139      }%
140      \global\@namedef{end#1}{\endtheorem}
141      \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
142  \fi
143 }
144
145 \gdef\@othm#1[#2]#3{%
146   \ifundefined{c@#2}{\@nocounterr{#2}}%
147   {\ifthm@tempif
148     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% lwarp
149     \global\@namedef{the#1}{\@nameuse{the#2}}%
150     \expandafter\protected@xdef\csname num@addtheorem#1\endcsname{%
151       \noexpand\@num@addtheorem#1}{#3}}%
152     \expandafter\protected@xdef\csname nonum@addtheorem#1\endcsname{%
153       \noexpand\@nonum@addtheorem#1}{#3}}%
154     \theoremkeyword{#3}%
155     \expandafter\protected@xdef\csname #1Keyword\endcsname
156       {\the\theoremkeyword}%
157     \expandafter\gdef\csname mkheader@#1\endcsname
158       {\csname setparms@#1\endcsname
159        \@thm{#1}{#2}{#3}
160       }%
161     \global\@namedef{end#1}{\endtheorem}
162     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
163   \fi}
164 }

```

§ 273.4 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.

```

165 \newcommand{\LWR@inctheorem}{%
166 \addtocounter{LWR@thisautoid}{1}%
167 \LWR@stoppars%
168 \LWR@htmltag{a id="\LWR@print@mbbox{autoid-\arabic{LWR@thisautoid}}"}\LWR@htmltag{/a}%
169 \LWR@startpars%
170 }

```

§ 273.5 \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>`.

```

171 \gdef\newtheoremstyle#1#2#3{%
172   \expandafter\@ifundefined{th@#1}%
173   {\expandafter\gdef\csname th@#1\endcsname{%
174     \def\@begintheorem####1####2{%
175       \LWR@intheorem% lwarp
176       #2}%
177     \def\@opargbegintheorem####1####2####3{%
178       \LWR@intheorem% lwarp
179       #3}%
180   }%
181 }%
182 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
183 }
```

§ 273.6 Standard styles

```

184 \renewtheoremstyle{plain}%
185   {\item[
186     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
187   {\item[
188     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
189
190 \renewtheoremstyle{break}%
191   {\item[
192     \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
193   ]}%
194   {\item[
195     \InlineClass{theoremheaderbreak}%
196       {##1\ ##2\ (##3)\theorem@separator}\newline
197   ]}
198
199 \renewtheoremstyle{change}%
200   {\item[
201     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
202   {\item[
203     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}
204
205 \renewtheoremstyle{changebreak}%
206   {\item[
207     \InlineClass{theoremheaderchangebreak}%
208       {##2\ ##1\theorem@separator}\newline
209   ]}%
210   {\item[
211     \InlineClass{theoremheaderchangebreak}%
212       {##2\ ##1\ (##3)\theorem@separator}\newline
213   ]}
214
```

```

215 \renewtheoremstyle{margin}%
216   {\item[
217     \InlineClass{theoremheadermargin}{##2 \quad ##1\theoremseparator}
218   ]}%
219   {\item[
220     \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theoremseparator}
221   ]}
222
223 \renewtheoremstyle{marginbreak}%
224   {\item[
225     \InlineClass{theoremheadermarginbreak}%
226       {##2 \quad ##1\theoremseparator}\newline
227   ]}%
228   {\item[
229     \InlineClass{theoremheadermarginbreak}%
230       {##2 \quad ##1\ (##3)\theoremseparator}\newline
231   ]}
232
233 \renewtheoremstyle{nonumberplain}%
234   {\item[
235     \InlineClass{theoremheaderplain}{##1\theoremseparator}]]}%
236   {\item[
237     \InlineClass{theoremheaderplain}{##1\ (##3)\theoremseparator}]]}
238
239 \renewtheoremstyle{nonumberbreak}%
240   {\item[
241     \InlineClass{theoremheaderbreak}{##1\theoremseparator}\newline
242   ]}%
243   {\item[
244     \InlineClass{theoremheaderbreak}{##1\ (##3)\theoremseparator}\newline
245   ]}
246
247 \renewtheoremstyle{empty}%
248   {\item[]}%
249   {\item[
250     \InlineClass{theoremheaderplain}{##3}]]}
251
252 \renewtheoremstyle{emptybreak}%
253   {\item[]}%
254   {\item[
255     \InlineClass{theoremheaderplain}{##3} \ \newline}

```

§ 273.7 Additional objects

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

256 \ifbool{LWR@theoremamsthm}{-}{%

```

Upright text via CSS:

```

257 \newtheoremstyle{plainupright}%
258 {\item[
259   \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]]}%
260 {\item[
261   \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]]}

```

Upright text and small caps header via CSS:

```

262 \newtheoremstyle{nonumberplainuprightsc}%
263 {\item[
264   \InlineClass{theoremheadersc}{##1\theorem@separator}]]}%
265 {\item[
266   \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]]}
267 }% not amsthm

```

§ 273.8 Renewed standard configuration

The following standard configuration is renewed using the new css:

```

268 \ifbool{LWR@ontheoremamsthm}{-}{%

269 \ifx\thm@usestd\@undefined
270 \else
271   \theoremnumbering{arabic}
272   \theoremstyle{plain}
273   \RequirePackage{latexsym}
274   \theoremsymbol{\Box}
275   \theorembodyfont{\itshape}
276   \theoremheaderfont{\normalfont\bfseries}
277   \theoremseparator{}
278   \renewtheorem{Theorem}{Theorem}
279   \renewtheorem{theorem}{Theorem}
280   \renewtheorem{Satz}{Satz}
281   \renewtheorem{satz}{Satz}
282   \renewtheorem{Proposition}{Proposition}
283   \renewtheorem{proposition}{Proposition}
284   \renewtheorem{Lemma}{Lemma}
285   \renewtheorem{lemma}{Lemma}
286   \renewtheorem{Korollar}{Korollar}
287   \renewtheorem{korollar}{Korollar}
288   \renewtheorem{Corollary}{Corollary}
289   \renewtheorem{corollary}{Corollary}
290
291   \theoremstyle{plainupright}
292   \theorembodyfont{\upshape}
293   \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
294   \renewtheorem{Example}{Example}

```

```

295 \renewtheorem{example}{Example}
296 \renewtheorem{Beispiel}{Beispiel}
297 \renewtheorem{beispiel}{Beispiel}
298 \renewtheorem{Bemerkung}{Bemerkung}
299 \renewtheorem{bemerkung}{Bemerkung}
300 \renewtheorem{Anmerkung}{Anmerkung}
301 \renewtheorem{anmerkung}{Anmerkung}
302 \renewtheorem{Remark}{Remark}
303 \renewtheorem{remark}{Remark}
304 \renewtheorem{Definition}{Definition}
305 \renewtheorem{definition}{Definition}
306
307 \theoremstyle{nonumberplainuprightsc}
308 \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
309 \renewtheorem{Proof}{Proof}
310 \renewtheorem{proof}{Proof}
311 \renewtheorem{Beweis}{Beweis}
312 \renewtheorem{beweis}{Beweis}
313 \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
314
315 \theoremsymbol{}
316 \fi
317 }% not amsthm

```

§ 273.9 amsthm option

Only if the amsthm option was given:

```

318 \ifbool{LWR@theoremamsthm}{
319
320 \gdef\th@plain{%
321 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
322 \def\@begintheorem##1##2{%
323 \LWR@intheorem% lwarp
324 \item[
325 \InlineClass{theoremheaderplain}{##1\ ##2.}
326 ]}%
327 \def\@opargbegintheorem##1##2##3{%
328 \LWR@intheorem% lwarp
329 \item[
330 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
331 ]}}
332
333 \gdef\th@nonumberplain{%
334 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
335 \def\@begintheorem##1##2{%
336 \LWR@intheorem% lwarp
337 \item[
338 \InlineClass{theoremheaderplain}{##1.}

```

```

339     ]}%
340 \def\@opargbegintheorem##1##2##3{%
341     \LWR@intheorem% lwarp
342     \item[
343 \InlineClass{theoremheaderplain}{##1\ (##3).}
344     ]}}
345
346 \gdef\th@definition{%
347 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
348 \def\@begintheorem##1##2{%
349     \LWR@intheorem% lwarp
350     \item[
351 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
352     ]}%
353 \def\@opargbegintheorem##1##2##3{%
354     \LWR@intheorem% lwarp
355     \item[
356 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
357     ]}}
358
359 \gdef\th@nonumberdefinition{%
360 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
361 \def\@begintheorem##1##2{%
362     \LWR@intheorem% lwarp
363     \item[
364 \InlineClass{theoremheaderdefinition}{##1.}
365     ]}%
366 \def\@opargbegintheorem##1##2##3{%
367     \LWR@intheorem% lwarp
368     \item[
369 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
370     ]}}
371
372 \gdef\th@remark{%
373 \def\theorem@headerfont{\itshape}\normalfont%
374 \def\@begintheorem##1##2{%
375     \LWR@intheorem% lwarp
376     \item[
377 \InlineClass{theoremheaderremark}{##1\ ##2.}
378     ]}%
379 \def\@opargbegintheorem##1##2##3{%
380     \LWR@intheorem% lwarp
381     \item[
382 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
383     ]}}
384
385 \gdef\th@nonumberremark{%
386 \def\theorem@headerfont{\itshape}\normalfont%
387 \def\@begintheorem##1##2{%
388     \LWR@intheorem% lwarp

```

```

389     \item[
390 \InlineClass{theoremheaderremark}{##1.}
391 ]}%
392 \def\@opargbegintheorem##1##2##3{%
393     \LWR@inctheorem% lwarp
394     \item[
395 \InlineClass{theoremheaderremark}{##1\ (##3).}
396 ]}]
397
398 \gdef\th@proof{%
399 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
400 \def\@begintheorem##1##2{%
401     \LWR@inctheorem% lwarp
402     \item[
403 \InlineClass{theoremheaderproof}{##1.}
404 ]}%
405 \def\@opargbegintheorem##1##2##3{%
406     \LWR@inctheorem% lwarp
407     \item[
408 \InlineClass{theoremheaderproof}{##1\ (##3).}
409 ]}]
410
411
412
413 \newcounter{proof}%
414 \if@thmmarks
415     \newcounter{currproofctr}%
416     \newcounter{endproofctr}%
417 \fi
418
419 \gdef\proofSymbol{\openbox}
420
421 \newcommand{\proofname}{Proof}
422
423 \newenvironment{proof}[1][\proofname]{
424     \th@proof
425     \def\theorem@headerfont{\itshape}%
426     \normalfont
427     \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
428     \@thm{proof}{proof}{#1}
429 }%
430 {\@endtheorem}
431
432 }{}% amsthm option

```

§ 273.10 Ending a theorem

Patched for CSS:

```

433 \let\LWR@origendtheorem\@endtheorem
434 \renewcommand{\@endtheorem}{%
435 \ifbool{LWR@ntheoremmarks}{%
436   \ifsetendmark%
437   \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
438   \setendmarkfalse%
439   \fi%
440 }{}%
441 \LWR@origendtheorem% also does \@endtrivlist
442 \ifbool{LWR@ntheoremmarks}{\global\setendmarktrue}{}%
443 \endBlockClass%
444 }
```

§ 273.11 \NoEndMark

```

445 \gdef\NoEndMark{\global\setendmarkfalse}
```

§ 273.12 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}
```

```

446 \renewcommand{\thm@@thmline@noname}[4]{%
447 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
448 }
449
450 \renewcommand{\thm@@thmline@name}[4]{%
451 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
452 }
```

This was redefined by **ntheorem** when loaded, so it is now redefined for **lwarp**:

```

453 \def\thm@@thmline{\thm@@thmline@name}
```

Patch for CSS:

```

454 \def\listtheorems#1{
455 \LWR@htmlclass{nav}{lothm}%
456 \begingroup
457 \c@tocdepth=-2%
458 \def\thm@list{#1}\thm@processlist
459 \endgroup
460 \LWR@htmlclassend{nav}{lothm}%
461 }
```

§ 273.13 Symbols

Proof QED symbol:

```

462 \newcommand{\qed}{\quad\the\qedsymbol}
463
464 \AtBeginDocument{
465   \@ifundefined{LWR@orig@openbox}{
466     \LetLtxMacro\LWR@orig@openbox\openbox
467     \LetLtxMacro\LWR@orig@blacksquare\blacksquare
468     \LetLtxMacro\LWR@orig@Box\Box
469
470     \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
471     \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
472     \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
473
474     \appto\LWR@restoreorigformatting{%
475       \LetLtxMacro\openbox\LWR@orig@openbox%
476       \LetLtxMacro\blacksquare\LWR@orig@blacksquare%
477       \LetLtxMacro\Box\LWR@orig@Box%
478     }% appto
479   }{}% \@ifundefined
480 }% \AtBeginDocument

```

§ 273.14 Cross-referencing

```

\thref {<label>}

481 \newcommand*{\thref}[1]{\cref{#1}}

```

File 182 **lwarp-overpic.sty**

§ 274 Package **overpic**

(Emulates or patches code by ROLF NIEPRASCHK.)

Pkg overpic **overpic** is patched for use by **lwarp**.



scaling

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

See section [80.2](#) for the print-mode version of `\overpicfontsize` and `\overpicfontskip`.

for HTML output: `1 \LWR@ProvidesPackagePass{overpic}`

File 183 lwarp-pagegrid.sty

Pkg	pagegrid	pagegrid is ignored.
-----	----------	-----------------------------

File 184 lwarp-pagenote.sty

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

for HTML output:

```
1 \DeclareOption{page}{}
2 \LWR@ProvidesPackagePass{pagenote}
```

File 185 **lwarp-pagesel.sty**

§ 277 Package **pagesel**

Pkg pagesel **pagesel** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pagesel}

File 186 **lwarp-paralist.sty**

§ 278 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 \LetLtxMacro\compactitem\itemize
3 \LetLtxMacro\compactenum\enumerate
4 \LetLtxMacro\compactdesc\description
5 \LetLtxMacro\endcompactitem\enditemize
6 \LetLtxMacro\endcompactenum\endenumerate
7 \LetLtxMacro\endcompactdesc\enddescription
```

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 187 **lwarp-parnotes.sty**

§ 279 Package **parnotes**
(Emulates or patches code by CHELSEA HUGHES.)

Pkg parnotes **parnotes** is supported with some patches.

for HTML output:

```

1 \LWR@ProvidesPackagePass{parnotes}

2 \long\def\PN@parnote@real#1#2{%
3   \parnotemark{#1}%
4   % Unless this is the first parnote in \PN@text, add a separator first
5   \unless\ifx\PN@text\@empty\g@addto@macro\PN@text{\parnoteintercmd}\fi
6   % Redefine \@currentlabel to the parnote label, so \label works
7   \g@addto@macro\PN@text{%
8     \phantomsection%
9     \def\@currentlabel{#1}%
10    \def\cref@currentlabel{%          lwarp
11      [parnotemark] [\arabic{parnotemark}] []\theparnotemark%
12    }%
13  }%
14  \g@addto@macro\PN@text{%
15    \LWR@textcurrentfont{%          lwarp
16      \parnotemark{#1}\nolinebreak\thinspace#2%
17    }%
18  }%
19 }
20
21 \def\PN@parnotes@real{%
22   % We call \par later, so this avoids recursion with \PN@parnotes@auto
23   \PN@inparnotesttrue
24   \unless\ifvmode\par\fi
25   % Avoid page breaks between a paragraph and its parnotes
26   \nopagebreak\addvspace{\parnotevskip}%
27   \LWR@forcenewpage%          lwarp
28   \begin{BlockClass}{footnotes}%          lwarp
29   {\parnotefmt{\PN@text}\par}%
30   \end{BlockClass}%          lwarp
31   \global\def\PN@text{}%
32   \addvspace{\parnotevskip}%
33   %
34   % These can be enabled or disabled by package options
35   %
36   \PN@disable@indent

```

```

37 \PN@reset@optional
38 \PN@inparnotesfalse
39 }
40
41 \AtBeginDocument{
42 \crefname{parnotemark}{paragraph note}{paragraph notes}
43 \Crefname{parnotemark}{Paragraph note}{Paragraph notes}
44 }

```

File 188 **lwarp-parskip.sty**

§ 280 Package **parskip**

Pkg parskip **parskip** is ignored.

for HTML output: Discard all options for **lwarp-parskip**.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

File 189 **lwarp-pbox.sty**

§ 281 Package **pbox**

(Emulates or patches code by SIMON LAW.)

Pkg pbox **pbox** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{pbox}

2 \NewDocumentCommand{\pbox}{0{t} o 0{t} m +m}{%
3 \booltrue{LWR@minipagefullwidth}%
4 \parbox[#1][#2][#3]{#4}{#5}%
5 }
6
7 \newcommand{\settominwidth}[3][\columnwidth]{%
8 \settowidth{#2}{#3}%
9 }
10
11 \newcommand{\widthhofpbox}[1]{%
12 \widthof{#1}%
13 }

```

File 190 **lwarp-pdfscape.sty**

§ 282 Package **pdfscape**

Pkg pdfscape **pdfscape** is ignored.

for HTML output: Discard all options for **lwarp-pdfscape**:

```
1 \LWR@ProvidesPackageDrop{pdfscape}
```

File 191 **lwarp-pdfrender.sty**

§ 283 Package **pdfrender**

Pkg pdfrender **pdfrender** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{pdfrender}

```
2 \newcommand*{\pdfrender}[1]{}
3 \newcommand{\textpdfrender}[2]{#2}
```

File 192 **lwarp-pdfsync.sty**

§ 284 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 193 **lwarp-pfnote.sty**

§ 285 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 194 **lwarp-phfqit.sty**

§ 286 Package **phfqit**

(Emulates or patches code by PHILIPPE FAIST.)

Pkg phfqit **phfqit** is patched for use by **lwarp**.

for HTML output: 1 \LWR@ProvidesPackagePass{phfqit}

```

2 \LetLtxMacro\LWR@origbitstring\bitstring
3
4 \renewcommand\bitstring[1]{%
5 \InlineClass[%
6     text-decoration: overline underline ;
7 ]{bitstring}{#1}%
8 % \phfqit@bitstring{#1}%
9 }
10
11 \appto\LWR@restoreorigformatting{%
12 \LetLtxMacro\bitstring\LWR@origbitstring%
13 }
```

File 195 **lwarp-placeins.sty**

§ 287 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 196 **lwarp-prelim2e.sty**

§ 288 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 197 **lwarp-prettyref.sty**

§ 289 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 198 **lwarp-preview.sty**

§ 290 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 199 **lwarp-quotchap.sty**

§ 291 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 200 **lwarp-quoting.sty**

§ 292 Package **quoting**

(Emulates or patches code by THOMAS TITZ.)

Pkg quoting **quoting** is patched for use by **lwarp**.

for HTML output:

```
1 \LWR@ProvidesPackagePass{quoting}

2 \xpatchcmd{\quoting}{\quo@begintext}
3   {\begin{LWR@blocktextcurrentfont}\quo@begintext}
4   {}
5   {
6     \PackageError{lwarp}
7       {Unable to patch the quoting package}
8       {Please contact the author of the lwarp package.}
9   }
10 \xpatchcmd{\endquoting}{\quo@endtext}
11   {\quo@endtext\end{LWR@blocktextcurrentfont}}
12   {}
13   {
14     \PackageError{lwarp}
15       {Unable to patch the quoting package}
16       {Please contact the author of the lwarp package.}
17   }
```

File 201 **lwarp-ragged2e.sty**

§ 293 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 \LetLtxMacro\Centering\centering
 3 \LetLtxMacro\RaggedLeft\raggedleft
 4 \LetLtxMacro\RaggedRight\raggedright
 5 \newcommand*{\justifying}{}\br/>
 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 202 **lwarp-realscripts.sty**

§ 294 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\realssubscript\textsubscript
4
5 \let\fakeuperscript\textsuperscript
6 \let\fakeubscript\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}}{%
15 \renewcommand{\LWR@realscriptsalign}{\LWR@print@mbox{text-align:center} ; }%
16 }{}%
17 \ifthenelse{\equal{#1}{r}}{%
18 \renewcommand{\LWR@realscriptsalign}{\LWR@print@mbox{text-align:right} ; }%
19 }{}%
20 }
21
22 \DeclareDocumentCommand \textsubsuperscript {s O{1} mm} {%
23 \LWR@setrealscriptsalign{#2}%
24 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
25 \textsuperscript{#4}\textsubscript{#3}%
26 }%
27 }
28
29 \DeclareDocumentCommand \textsupersubscript {s O{1} mm} {%
30 \LWR@setrealscriptsalign{#2}%
31 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
32 \textsubscript{#4}\textsuperscript{#3}%
33 }%
34 }

```

File 203 **lwarp-register.sty**

§ 295 Package **register**

(Emulates or patches code by MATTHEW LOVELL.)

Pkg register **register** is patched for use by **lwarp**.

```

for HTML output: 1 \LWR@ProvidesPackagePass{register}

```

```

2 \xpatchcmd{\register}
3   {\centering}
4   {\begin{center}\begin{lateximage}[-register~\packagediagramname]}
5   {\typeout{Patch register success.}}
6   {\typeout{Patch register failure.}}
7
8 \xpatchcmd{\endregister}
9   {\leftskip}
10  {%
11      \end{lateximage}\end{center}}%
12      \leftskip%
13  }%
14  {\typeout{Patch endregister success.}}
15  {\typeout{Patch endregister failure.}}
16
17 \setlength{\regWidth}{5in}

```

File 204 **lwarp-relsize.sty**

§ 296 Package **relsize**


(Emulates or patches code by DONALD ARSENEAU, BERNIE COSELL, MATT SWIFT.)

Pkg relsize **relsize** is patched for use by **lwarp**.

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

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

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

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

for HTML output:

```

1 \LWR@ProvidesPackagePass{relsize}

2 \let\LWR@origrelsize\relsize
3 \LetLtxMacro\LWR@origlarger\larger
4 \LetLtxMacro\LWR@origsmaller\smaller
5 \let\LWR@relscale\relscale
6 \LetLtxMacro\LWR@origtextlarger\textlarger
7 \LetLtxMacro\LWR@origtextsmaller\textsmaller
8 \let\LWR@textscale\textscale
9
10 \appto\LWR@restoreorigformatting{%

```

```

11 \let\relsize\LWR@origrelsize%
12 \LetLtxMacro\larger\LWR@origlarger%
13 \LetLtxMacro\smaller\LWR@origsmaller%
14 \let\relscale\LWR@relscale%
15 \LetLtxMacro\textlarger\LWR@origtextlarger%
16 \LetLtxMacro\textsmaller\LWR@origtextsmaller%
17 \let\textscale\LWR@textscale%
18 }
19
20 \newcounter{LWR@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 205 **lwarp-resizegather.sty**

§ 297 Package **resizegather**

Pkg **resizegather** **resizegather** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{resizegather}

2 \newcommand*{\resizegathersetup}[1]{%

```

File 206 **lwarp-romanbar.sty**

§ 298 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 207 **lwarp-romanbarpagenumber.sty**

§ 299 Package **romanbarpagenumber**

Pkg romanbarpagenumber **romanbarpagenumber** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{romanbarpagenumber}
```

File 208 **lwarp-rotating.sty**

§ 300 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 209 **lwarp-rotfloat.sty**

§ 301 Package **rotfloat**

(Emulates or patches code by AXEL SOMMERFELDT.)

Pkg rotfloat **rotfloat** is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotfloat}[2004/01/04]
2
3 \RequirePackage{float}

```

`\newfloat` $\{ \langle 1: type \rangle \} \{ \langle 2: placement \rangle \} \{ \langle 3: ext \rangle \} [\langle 4: within \rangle]$

Emulates the `\newfloat` command from the **float** package. Sideways floats are `\let` to the same as regular floats.

“placement” is ignored.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}%
6 {%
7   \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}%
8 }%
9 {%
10  \DeclareFloatingEnvironment[fileext=#3]{#1}%
11 }%
12 \csletcs{sideways#1}{#1}%
13 \csletcs{endsideways#1}{end#1}%

```

Remember the float style:

```

14 \csedef{LWR@floatstyle@#1}{\LWR@floatstyle}%
15 \csedef{LWR@floatstyle@sideways#1}{\LWR@floatstyle}%

```

newfloat package automatically creates the `\listof` command for new floats, but **float** does not, so remove `\listof` here in case it is manually created later:

```
16 \cslet{listof#1s}\relax%
17 \cslet{listof#1es}\relax%
18 \cslet{listofsideways#1s}\relax%
19 \cslet{listofsideways#1es}\relax%
20 }
```

File 210 **lwarp-savetrees.sty**

§ 302 Package **savetrees**

Pkg **savetrees** Emulated.

for HTML output: Discard all options for **lwarp-savetrees**:

```
1 \LWR@ProvidesPackageDrop{savetrees}
```

File 211 **lwarp-scalefnt.sty**

§ 303 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 212 **lwarp-schemata.sty**

§ 304 Package **schemata**

(Emulates or patches code by CHARLES P. SCHAUUM.)

Pkg **schemata** **schemata** is patched for use by **lwarp**.

for HTML output: 1 \LWR@ProvidesPackagePass{schemata}

```

2 \LetLtxMacro\LWR@schemata@origschema\schem
3 \LetLtxMacro\LWR@schemata@origSchema\Schema
4
5 \renewcommand{\schema}[3][open]{%
6 \begin{lateximage}%
7 \LWR@orignormalsize
8 \LWR@schemata@origschema[#1]{#2}{#3}%
9 \end{lateximage}%
10 }
11
12 \renewcommand{\Schema}[5][open]{%
13 \begin{lateximage}%
14 \LWR@orignormalsize
15 \LWR@schemata@origSchema[#1]{#2}{#3}{#4}{#5}%
16 \end{lateximage}%
17 }

```

File 213 **lwarp-scrextend.sty**

\$ 305 Package **scrextend**

Pkg scrextend **scrextend** is emulated.

This package may be loaded standalone, but is also loaded automatically if **koma-script** classes are in use. `\DeclareDocumentCommand` is used to overwrite the **koma-script** definitions.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{scrextend}

2 \DeclareDocumentCommand{\setkomafont}{m m}{}
3 \DeclareDocumentCommand{\addkomafont}{m m}{}
4 \DeclareDocumentCommand{\usekomafont}{m}{}
5
6 \DeclareDocumentCommand{\usefontofkomafont}{m}{}
7 \DeclareDocumentCommand{\useencodingofkomafont}{m}{}
8 \DeclareDocumentCommand{\usesizeofkomafont}{m}{}
9 \DeclareDocumentCommand{\usefamilyofkomafont}{m}{}
10 \DeclareDocumentCommand{\useseriesofkomafont}{m}{}
11 \DeclareDocumentCommand{\useshapeofkomafont}{m}{}
12
13 \AtBeginDocument{
14 \let\LWR@maketitle\maketitle
15 \DeclareDocumentCommand{\maketitle}{o}{\LWR@maketitle}
16 }
17
18 \DeclareDocumentCommand{\extratitle}{m}{}
19 \DeclareDocumentCommand{\titlehead}{m}{}

```

```

20 \DeclareDocumentCommand{\subject}{m}{}
21 \DeclareDocumentCommand{\publishers}{m}{\published{#1}}
22 \DeclareDocumentCommand{\uppertitleback}{m}{}
23 \DeclareDocumentCommand{\lowertitleback}{m}{}
24 \DeclareDocumentCommand{\dedication}{m}{}
25
26 \DeclareDocumentCommand{\ifthispageodd}{m m}{#1}
27
28 \DeclareDocumentCommand{\titlepagestyle}{}{}
29
30 \DeclareDocumentCommand{\cleardoublepageusingstyle}{m}{}
31 \DeclareDocumentCommand{\cleardoubleemptypage}{}{}
32 \DeclareDocumentCommand{\cleardoubleplainpage}{}{}
33 \DeclareDocumentCommand{\cleardoublestandardpage}{}{}
34 \DeclareDocumentCommand{\cleardoubleoddpge}{}{}
35 \DeclareDocumentCommand{\cleardoubleoddpgeusingstyle}{m}{}
36 \DeclareDocumentCommand{\cleardoubleoddpgeemptypage}{}{}
37 \DeclareDocumentCommand{\cleardoubleoddpgeplainpage}{}{}
38 \DeclareDocumentCommand{\cleardoubleoddpgestandardpage}{}{}
39 \DeclareDocumentCommand{\cleardoubleevenpage}{}{}
40 \DeclareDocumentCommand{\cleardoubleevenpageusingstyle}{m}{}
41 \DeclareDocumentCommand{\cleardoubleevenemptypage}{}{}
42 \DeclareDocumentCommand{\cleardoubleevenplainpage}{}{}
43 \DeclareDocumentCommand{\cleardoubleevenstandardpage}{}{}
44
45 \DeclareDocumentCommand{\multiplefootnoteseparator}{}{%
46   \begingroup\let\thefootnotemark\multfootsep\makefnmark\endgroup
47 }
48
49 \DeclareDocumentCommand{\multfootsep}{}{,}
50
51 \DeclareDocumentCommand{\footref}{m}{%
52   \begingroup
53     \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
54   \endgroup
55   \@footnotemark
56 }
57
58 \DeclareDocumentCommand{\deffootnote}{o m m m}{}
59 \DeclareDocumentCommand{\deffootnotemark}{m}{}
60 \DeclareDocumentCommand{\setfootnoterule}{o m}{}
61 \DeclareDocumentCommand{\raggedfootnote}{}{}
62
63 \DeclareDocumentCommand{\dictum}{o m}{
64   \begin{LWR@BlockClassWP}{\LWR@print@mbx{text-align:right}}{}{\dictum}
65     #2
66     \IfValueT{#1}
67     {
68       \ifbool{FormatWP}
69       {\begin{BlockClass}[\LWR@print@mbx{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 \qqquad \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@print@mbox{margin-left:\LWR@printlength{\LWR@templengthtwo}} ;
110     \LWR@print@mbox{margin-right:\LWR@printlength{\LWR@templengthtwo}}
111   ]{addmargin}
112 }
113 {
114   \setlength{\LWR@templengthone}{#2}
115   \begin{BlockClass}[
116     \LWR@print@mbox{margin-left:\LWR@printlength{\LWR@templengthone}} ;
117     \LWR@print@mbox{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 214 **lwarp-scrhack.sty**

§ 306 Package **scrhack**

Pkg scrhack **scrhack** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{scrhack}

File 215 **lwarp-scrlayer.sty**

§ 307 Package **scrlayer**

(Emulates or patches code by MARKUS KOHM.)

Pkg scrlayer **scrlayer** is emulated.

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

for HTML output: 1 \LWR@ProvidesPackageDrop{scrlayer}

```

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*{\layerhaligh}{}
9 \newcommand*{\layervaligh}{}

```

```

10 \newcommand*{\layerxoffset}{}
11 \newcommand*{\laveryoffset}{}
12 \newcommand*{\layerwidth}{}
13 \newcommand*{\layerheight}{}
14 \providecommand*{\LenToUnit}[1]{\strip@pt\dimexpr#1*\p@/\unitlength}
15 \newcommand*{\putUL}[1]{}
16 \newcommand*{\putUR}[1]{}
17 \newcommand*{\putLL}[1]{}
18 \newcommand*{\putLR}[1]{}
19 \newcommand*{\putC}[1]{}
20 \newcommand*{\GetLayerContents}[1]{}
21 \newcommand*{\IfLayerExists}[3]{#3}
22 \newcommand*{\DestroyLayer}[1]{}
23 \newcommand*{\layercontentsmeasure}{}
24 \newcommand*{\currentpagestyle}{}
25 \newcommand*{\BeforeSelectAnyPageStyle}[1]{}
26 \newcommand*{\AfterSelectAnyPageStyle}[1]{}
27 \newcommand*{\DeclarePageStyleAlias}[2]{}
28 \newcommand*{\DeclareNewPageStyleAlias}[2]{}
29 \newcommand*{\ProvidePageStyleAlias}[2]{}
30 \newcommand*{\RedeclarePageStyleAlias}[2]{}
31 \newcommand*{\DestroyPageStyleAlias}[1]{}
32 \newcommand*{\GetRealPageStyle}[1]{}
33 \newcommand*{\DeclarePageStyleByLayers}[3][{}]{ }
34 \newcommand*{\DeclareNewPageStyleByLayers}[3][{}]{ }
35 \newcommand*{\ProvidePageStyleByLayers}[3][{}]{ }
36 \newcommand*{\RedeclarePageStyleByLayers}[3][{}]{ }
37 \NewDocumentCommand{\ForEachLayerOfPageStyle}{s m m}{}
38 \newcommand*{\AddLayersToPageStyle}[2]{}
39 \newcommand*{\AddLayersAtBeginOfPageStyle}[2]{}
40 \newcommand*{\AddLayersAtEndOfPageStyle}[2]{}
41 \newcommand*{\RemoveLayersFromPageStyle}[2]{}
42 \newcommand*{\AddLayersToPageStyleBeforeLayer}[3]{}
43 \newcommand*{\AddLayersToPageStyleAfterLayer}[3]{}
44 \newcommand*{\UnifyLayersAtPageStyle}[1]{}
45 \newcommand*{\ModifyLayerPageStyleOptions}[2]{}
46 \newcommand*{\AddToLayerPageStyleOptions}[2]{}
47 \newcommand*{\IfLayerPageStyleExists}[3]{#3}
48 \newcommand*{\IfRealLayerPageStyleExists}[3]{#3}
49 \newcommand*{\IfLayerAtPageStyle}[4]{#4}
50 \newcommand*{\IfSomeLayerAtPageStyle}[4]{#4}
51 \newcommand*{\IfLayersAtPageStyle}[4]{#4}
52 \newcommand*{\DestroyRealLayerPageStyle}[1]{}
53 \@ifundefined{footheight}{\newlength{footheight}}{}
54 \DeclareDocumentCommand{\automark}{s o m}{}
55 \DeclareDocumentCommand{\manualmark}{}{}
56 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
57 \DeclareDocumentCommand{\GenericMarkFormat}{}{}
58 \newcommand*{\@mkleft}[1]{}
59 \newcommand*{\@mkright}[1]{}

```

```

60 \newcommand*{\@mkdouble}[1]{}
61 \newcommand*{\@mkboth}[2]{}
62 \newcommand*{\sclayerInitInterface}[1] [] {}
63 \newcommand*{\sclayerAddToInterface}[3] [] {}
64 \newcommand*{\sclayerAddCsToInterface}[3] [] {}
65 \newcommand*{\sclayerOnAutoRemoveInterface}[2] [] {}

```

File 216 **lwarp-sclayer-notecolumn.sty**

§ 308 Package **sclayer-notecolumn**

(Emulates or patches code by MARKUS KOHM.)

Pkg sclayer-notecolumn **sclayer-notecolumn** is emulated.

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

for HTML output:

```

1 \LWR@ProvidesPackageDrop{sclayer-notecolumn}

2 \newcommand*{\DeclareNoteColumn}[2] [] {}
3 \newcommand*{\DeclareNewNoteColumn}[2] [] {}
4 \newcommand*{\ProvideNoteColumn}[2] [] {}
5 \newcommand*{\RedeclareNoteColumn}[2] [] {}
6 \NewDocumentCommand{\makenote}{s o m}{\marginpar{#3}}
7 \newcommand*{\syncwithnotecolumn}[1] [] {}
8 \newcommand*{\syncwithnotecolumns}[1] [] {}
9 \newcommand*{\clearnotecolumn}[1] [] {}
10 \newcommand*{\clearnotecolumns}[1] [] {}

```

File 217 **lwarp-sclayer-scrpage.sty**

§ 309 Package **sclayer-scrpage**

(Emulates or patches code by MARKUS KOHM.)

Pkg sclayer-scrpage **sclayer-scrpage** is emulated.

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

for HTML output:

```

1 \LWR@ProvidesPackageDrop{sclayer-scrpage}

2 \@ifundefined{footheight}{\newlength{footheight}}{}
3 \NewDocumentCommand{\lehead}{s o m}{}
4 \NewDocumentCommand{\cehead}{s o m}{}

```

```

5 \NewDocumentCommand{\rehead}{s o m}{}
6 \NewDocumentCommand{\lohead}{s o m}{}
7 \NewDocumentCommand{\cohead}{s o m}{}
8 \NewDocumentCommand{\rohead}{s o m}{}
9 \NewDocumentCommand{\lefoot}{s o m}{}
10 \NewDocumentCommand{\cefoot}{s o m}{}
11 \NewDocumentCommand{\refoot}{s o m}{}
12 \NewDocumentCommand{\lofoot}{s o m}{}
13 \NewDocumentCommand{\cofoot}{s o m}{}
14 \NewDocumentCommand{\rofoot}{s o m}{}
15 \NewDocumentCommand{\ohead}{s o m}{}
16 \NewDocumentCommand{\chead}{s o m}{}
17 \NewDocumentCommand{\ihead}{s o m}{}
18 \NewDocumentCommand{\ofoot}{s o m}{}
19 \NewDocumentCommand{\cfoot}{s o m}{}
20 \NewDocumentCommand{\ifoot}{s o m}{}
21 \DeclareDocumentCommand{\automark}{s o m}{}
22 \DeclareDocumentCommand{\manualmark}{}{}
23 \DeclareDocumentCommand{\MakeMarkcase}{m}{#1}
24 \DeclareDocumentCommand{\GenericMarkFormat}{}{}
25 \newcommand*{\defpairofpagestyles}[3] [] {}
26 \newcommand*{\newpairofpagestyles}[3] [] {}
27 \newcommand*{\renewpairofpagestyles}[3] [] {}
28 \newcommand*{\providepairofpagestyles}[3] [] {}
29 \newcommand*{\clearmainofpairofpagestyles}
30 \newcommand*{\clearplainofpairofpagestyles}
31 \newcommand*{\clearpairofpagestyles}
32 \NewDocumentCommand{\deftriplepagestyle}{m o o m m m m m m}{}
33 \NewDocumentCommand{\newtriplepagestyle}{m o o m m m m m m}{}
34 \NewDocumentCommand{\renewtriplepagestyle}{m o o m m m m m m}{}
35 \NewDocumentCommand{\providetriplespagestyle}{m o o m m m m m m}{}
36 \newcommand*{\defpagestyle}[3] {}
37 \newcommand*{\newpagestyle}[3] {}
38 \newcommand*{\providepagestyle}[3] {}
39 \newcommand*{\renewpagestyle}[3] {}

```

File 218 **lwarp-section.sty**

§ 310 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\ttsi\LARGE
7 \let\ausi\large
8 \let\dasizelarge
9 \let\secsiLarge
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 219 **lwarp-sectionbreak.sty**

§ 311 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 220 **lwarp-sectsty.sty**

§ 312 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 221 **lwarp-setspace.sty**

§ 313 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 222 **lwarp-shadow.sty**

§ 314 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 223 **lwarp-showidx.sty**

§ 315 Package **showidx**

Pkg showidx **showidx** is ignored.

for HTML output: Discard all options for **lwarp-showidx**:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

File 224 **lwarp-showkeys.sty**

§ 316 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 225 **lwarp-sidecap.sty**

§ 317 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 226 **lwarp-sidenotes.sty**

§ 318 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{KFLTsidefloat}{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 \TeX 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 227 **lwarp-siunitx.sty**

§ 319 Package **siunitx**


(Emulates or patches code by JOSEPH WRIGHT.)

Pkg **siunitx** **siunitx** is patched for use by **lwarp**.

fractions Due to **pdf_latex** limitations, fraction output is replaced by symbol output for per-mode and quotient-mode.

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

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

 **tabular** Tabular S columns are rendered as simple c columns, and tabular s columns are not supported. These may be replaced by c columns with each cell contained in `\num` or `\si`.

for HTML output:

```

1 \RequirePackage{xcolor}% for \convertcolorspec
2
3 \LWR@ProvidesPackagePass{siunitx}

4 \AtBeginDocument{% in case textcomp was not loaded
5 \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
6 \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
7 \DeclareSIUnit\elementarycharge{\textit{e}}
8 \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
9 \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
10 \DeclareSIUnit\planckbar{\LWR@siunitx@textplanckbar}
11}% 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.

```

12
13 \ExplSyntaxOn
14 %

```

Modified to set HTML \textcolor if not black:

```

15 \cs_undefine:N \__siunitx_print_aux:
16 \cs_new_protected:Npn \__siunitx_print_aux:
17 {
18   \text
19   {
20     \__siunitx_ensure_ltr:n
21     {
22       \color@begingroup
23       \__siunitx_print_color:
24       \__siunitx_font_shape:
25       \__siunitx_font_weight:
26       \use:c
27       {
28         @@_ \l__siunitx_print_type_tl _
29         text \l__siunitx_font_family_tl :
30       }
31       \bool_if:NTF \l__siunitx_font_math_mode_bool
32       { \__siunitx_print_math: }
33       {
34         \LWR@findcurrenttextcolor% lwarp
35         \ifdefstring{\LWR@tempcolor}{000000}% lwarp
36         {\__siunitx_print_text:}% lwarp
37         {% lwarp
38           \LWR@textcurrentcolor{% lwarp
39             \__siunitx_print_text:

```

```

40          }% lwarp
41      }% lwarp
42  }
43  \color@endgroup
44  }
45  }
46  }
47
48
49 \cs_undefine:N \__siunitx_set_math_fam:n
50 \cs_new_protected:Npn \__siunitx_set_math_fam:n #1 {
51   \int_new:c { c__siunitx_math #1 _int }
52   \group_begin:% lwarp
53     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
54     \LetLtxMacro\mbox\LWR@print@mbox% lwarp
55     \hbox_set:Nn \l__siunitx_tmp_box
56     {
57       \ensuremath
58       {
59         \use:c { math #1 }
60         {
61           \int_gset:cn { c__siunitx_math #1 _int } { \fam }
62         }
63       }
64     }
65   \group_end:% lwarp
66 }
67
68 \cs_undefine:N \__siunitx_combined_output:n
69 \cs_new_protected:Npn \__siunitx_combined_output:n #1 {
70   \group_begin:% lwarp
71   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
72   \LetLtxMacro\mbox\LWR@print@mbox% lwarp
73   \bool_if:NTF \l__siunitx_number_parse_bool
74   {
75     \tl_clear:N \l__siunitx_number_out_tl
76     \bool_set_false:N \l__siunitx_number_compound_bool
77     \__siunitx_number_output_parse:n {#1}
78   }
79   {
80     \__siunitx_unit_output_pre_print:

```

For parse-numbers=false:

```

81 %     \__siunitx_print:nn { number } { \ensuremath {#1} }
82     \LWR@subsingledollar{% lwarp
83       \textbackslash( \LWR@HTMLsanitize{#1} \textbackslash)% lwarp
84     }{siunitx}{%
85       \__siunitx_print:nn { number } {%

```

```

86          \LWR@origensuredmath{#1}%
87      }%
88  }% lwarp

89  \__siunitx_unit_output_print:
90  }
91  \group_end:% lwarp
92 }
93 %

```

For quotients, the fraction code is replaced by the symbol code:

```

94 \cs_undefine:N \__siunitx_number_output_quotient_fraction:
95 \cs_new_protected:Npn \__siunitx_number_output_quotient_fraction: {
96   \bool_set_true:N \l__siunitx_number_compound_bool
97   \__siunitx_number_output_quotient_aux_i:
98   \tl_set_eq:NN \l__siunitx_number_out_tl
99     \l__siunitx_number_numerator_tl
100   \tl_put_right:NV \l__siunitx_number_out_tl \l__siunitx_output_quotient_tl
101   \tl_put_right:NV \l__siunitx_number_out_tl
102     \l__siunitx_number_denominator_tl
103   \__siunitx_number_output_single_aux:
104 }

```

For units, the fraction code is replaced by the symbol code:

```

105 \cs_undefine:N \__siunitx_unit_format_fraction_fraction:
106 \cs_new_protected:Npn \__siunitx_unit_format_fraction_fraction: {
107   \__siunitx_unit_format_fraction_symbol_aux:
108   \int_compare:nNnT { \l__siunitx_unit_denominator_int } > { 1 }
109   {
110     \bool_if:NT \l__siunitx_unit_denominator_bracket_bool
111     {
112       \tl_put_left:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_open_tl
113       \tl_put_right:NV \l__siunitx_unit_denominator_tl \l__siunitx_bracket_close_tl
114     }
115   }
116   \tl_set_eq:NN \l__siunitx_unit_tl \l__siunitx_unit_numerator_tl
117   \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_per_symbol_tl
118   \tl_put_right:NV \l__siunitx_unit_tl \l__siunitx_unit_denominator_tl
119 }

120 \cs_undefine:N \__siunitx_angle_print_astronomy_aux:
121 \cs_new_protected:Npn \__siunitx_angle_print_astronomy_aux: {
122   \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-integer }
123     \l__siunitx_tmpa_tl
124   { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
125   \ifnumcomp{\value{LWR@lateximagedepth}}{>}{0}% lwarp

```

```

126 {% lateximage
127   \hbox_set:Nn \l__siunitx_angle_marker_box
128   {
129     \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
130   }
131   \hbox_set:Nn \l__siunitx_angle_unit_box
132   {
133     \__siunitx_print:nV { unit } \l__siunitx_unit_tl
134     \skip_horizontal:n { -\scriptspace }
135   }
136   \__siunitx_angle_print_astronomy_aux:n { marker }
137   \__siunitx_angle_print_astronomy_aux:n { unit }
138   \hbox_set:Nn \l__siunitx_angle_marker_box
139   {
140     \box_use:N \l__siunitx_angle_marker_box
141     \box_use:N \l__siunitx_angle_unit_box
142   }
143   \dim_compare:nNnTF
144     { \l__siunitx_angle_marker_dim } > { \l__siunitx_angle_unit_dim }
145     { \__siunitx_angle_print_astronomy_marker: }
146     { \__siunitx_angle_print_astronomy_unit: }
147 }% lateximage
148 {% not a lateximage
149   \__siunitx_print:nV { unit } \l__siunitx_unit_tl
150   \__siunitx_print:nn { number } { { \l__siunitx_output_decimal_tl } }
151 }% not a lateximage
152 \prop_get:NnNT \l__siunitx_number_out_prop { mantissa-decimal }
153   \l__siunitx_tmpa_tl
154   { \__siunitx_print:nV { number } \l__siunitx_tmpa_tl }
155 }

156 \RenewDocumentCommand \num { o m } {
157   \leavevmode
158   \group_begin:% lwarp
159     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
160     \LetLtxMacro\mbox\LWR@print@mbox% lwarp
161     \bool_set_false:N \l__siunitx_font_set_bool
162     \IfNoValueF {#1}
163       { \keys_set:nn { siunitx } {#1} }
164     \__siunitx_number_output:n {#2}
165   \group_end:% lwarp
166 }

167
168 \RenewDocumentCommand \numrange { o m m } {
169   \leavevmode
170   \group_begin:% lwarp
171     \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
172     \LetLtxMacro\mbox\LWR@print@mbox% lwarp
173     \bool_set_false:N \l__siunitx_font_set_bool

```

```

174 \IfNoValueF {#1}
175   { \keys_set:nn { siunitx } {#1} }
176   \_siunitx_range_numbers:nn {#2} {#3}
177 \group_end:% lwarp
178 }
179
180 \RenewDocumentCommand \ang { o > { \SplitArgument { 2 } { ; } } m } {
181   \group_begin:% lwarp
182   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
183   \LetLtxMacro\mbox\LWR@print@mbox% lwarp
184   \IfNoValueF {#1}
185     { \keys_set:nn { siunitx } {#1} }
186     \_siunitx_angle_output:nnn #2
187   \group_end:% lwarp
188 }
189
190 \RenewDocumentCommand \si { o m } {
191   \leavevmode
192   \group_begin:% lwarp
193   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
194   \LetLtxMacro\mbox\LWR@print@mbox% lwarp
195   \bool_set_false:N \l__siunitx_font_set_bool
196   \IfNoValueTF {#1}
197     { \_siunitx_unit_output:nn {#2} { } }
198     {
199       \keys_set:nn { siunitx } {#1}
200       \_siunitx_unit_output:nn {#2} {#1}
201     }
202   \group_end:% lwarp
203 }
204
205
206 \RenewDocumentCommand{\SIrange}{o m m m}
207 {%
208   \leavevmode
209   \group_begin:% lwarp
210   \LetLtxMacro\@ensuredmath\LWR@origensuredmath% lwarp
211   \LetLtxMacro\mbox\LWR@print@mbox% lwarp
212   \bool_set_false:N \l__siunitx_font_set_bool
213   \IfNoValueTF {#1}
214     { \_siunitx_range_unit:nnnn {#4} { } {#2} {#3} }
215     {
216       \keys_set:nn { siunitx } {#1}
217       \_siunitx_range_unit:nnnn {#4} {#1} {#2} {#3}
218     }
219   \group_end:% lwarp
220 }
221
222 \ExplSyntaxOff

```

File 228 **lwarp-soul.sty**

§ 320 Package **soul**

(Emulates or patches code by MELCHIOR FRANZ.)

Pkg soul Emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{soul}[2003/11/17]
 2 \RequirePackage{xcolor}% for \convertcolorspec

Storage for the colors to use:

```
3 \newcommand*\LWR@soululcolor{}
4
5 \newcommand*\LWR@soulstcolor{}
6
7 % \definecolor{\LWR@soulhlcolordefault}{HTML}{F8E800}
8 % \newcommand*\LWR@soulhlcolor{\LWR@soulhlcolordefault}
9 \newcommand*\LWR@soulhlcolor{}
```

\so {<text>}

Basic markup with css:

```
10 \newcommand{\so}[1]{%
11 \LWR@HTMLtextstyle{letter-spacing:.2ex}{letterspacing}{#1}%
12 }
```

\caps {<text>}

```
13 \newcommand{\caps}[1]{%
14 \LWR@HTMLtextstyle%
15     {font-variant:small-caps;letter-spacing:.1ex}%
16     {capsspacing}{#1}%
17 }
```

\LWR@soulcolor {<text>} {<color>} {<class>} {<colorstyle>} {<FormatWPstyle>}

Add colors if not empty:

```
18 \newcommand{\LWR@soulcolor}[5]{%
19 \ifcsempy{#2}%
20 {\LWR@HTMLtextstyle{#5}{#3}{#1}}%
21 {%
```

```

22 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
23 \LWR@htmlspanclass[#5;#4:\LWR@origpound\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: auto;}%
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:\LWR@origpound{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 229 **lwarp-soulpos.sty**

§ 321 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 230 **lwarp-soulutf8.sty**

§ 322 Package **soulutf8**

Pkg soulutf8 **soulutf8** is emulated.

lwarp's HTML output naturally supports UTF-8 encoding.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{soulutf8}

```

File 231 **lwarp-stabular.sty**

§ 323 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 232 **lwarp-stfloats.sty**

§ 324 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 233 **lwarp-subfig.sty**

§ 325 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` `{\langle 1 type \rangle} [\langle 2 lof entry \rangle] [\langle 3 caption \rangle] {\langle 4 contents \rangle}`

The outer minipage allows side-by-side subfloats with `\hfill` between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% 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 234 **lwarp-subfigure.sty**

§ 326 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\ifsubfiguretopcap
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 235 **lwarp-supertabular.sty**

§ 327 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
...
\StopDefiningTabulars
```

See section 9.9.

⚠ 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   \LWR@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 236 **lwarp-syntonly.sty**

§ 328 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*{\syntaxonly}{}
6
7 \@onlypreamble\syntaxonly

```

File 237 **lwarp-t1enc.sty**

§ 329 Package **t1enc**

Pkg t1enc **t1enc** does not work with **lwarp**.

for HTML output: 1 \LWR@loadnever{t1enc}{fontenc, inputenc, inputenx}

File 238 **lwarp-tables.sty**

§ 330 Package **tables**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg tables **tables** is emulated. \LWR@hline is used to handle the optional argument when **tables** is loaded.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{tables}

2 \newdimen\tablinesep
3 \newdimen\arraylinesep
4 \newdimen\extrarulesep
```

File 239 **lwarp-tabularx.sty**

§ 331 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 240 **lwarp-tabulary.sty**

§ 332 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 241 **lwarp-textarea.sty**

§ 333 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 242 **lwarp-textcomp.sty**

§ 334 Package **textcomp**

(Emulates or patches code by FRANK MITTELBACH, ROBIN FAIRBAIRNS, WERNER LEMBERG.)

Pkg `textcomp` **textcomp** is patched for use by **lwarp**.

§ 334.1 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.

§ 334.2 Package loading

for HTML output: `1 \LWR@ProvidesPackagePass{textcomp}`

§ 334.3 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@origtextthreequartersemdash\textthreequartersemdash
11 \let\LWR@origtextmho\textmho
12 \let\LWR@origtextnaira\textnaira
13 \let\LWR@origtextpeso\textpeso
14 \let\LWR@origtextrecipe\textrecipe
15 \let\LWR@origtextinterrobangdown\textinterrobangdown
16 \let\LWR@origtextperthousand\textperthousand
17 \let\LWR@origtextpertenthousand\textpertenthousand
18 \let\LWR@origtextbaht\textbaht
19 \let\LWR@origtextdiscount\textdiscount
20 \let\LWR@origtextservicemark\textservicemark
21 \LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
22 \LetLtxMacro\LWR@origcapitalogonek\capitalogonek
23 \LetLtxMacro\LWR@origcapitalgrave\capitalgrave
24 \LetLtxMacro\LWR@origcapitalacute\capitalacute
25 \LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex
26 \LetLtxMacro\LWR@origcapitaltilde\capitaltilde
27 \LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
28 \LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
29 \LetLtxMacro\LWR@origcapitalring\capitalring
30 \LetLtxMacro\LWR@origcapitalcaron\capitalcaron
31 \LetLtxMacro\LWR@origcapitalbreve\capitalbreve
32 \LetLtxMacro\LWR@origcapitalmacron\capitalmacron
33 \LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
34 \LetLtxMacro\LWR@origtextcircled\textcircled

```

§ 334.4 HTML symbols

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for Lua \TeX and Xe \TeX .

§ 334.4.1 pdf \TeX symbols

```

35 \AtBeginDocument{
36 \ifPDFTeX
37 \renewcommand*{\textdegree}{\HTMLentity{deg}}
38 \renewcommand*{\textcelsius}{\HTMLUnicode{2103}}
39 \renewcommand*{\textohm}{\HTMLUnicode{2126}}
40 \renewcommand*{\textmu}{\HTMLUnicode{00B5}}
41 \renewcommand*{\textlquill}{\HTMLUnicode{2045}}
42 \renewcommand*{\textrquill}{\HTMLUnicode{2046}}
43 \renewcommand*{\textcircledP}{\HTMLUnicode{2117}}
44 \renewcommand*{\texttwelveudash}{\HTMLUnicode{2014}}% emdash
45 \renewcommand*{\textthreequartersemdash}{\HTMLUnicode{2014}}% emdash
46 \renewcommand*{\textmho}{\HTMLUnicode{2127}}
47 \renewcommand*{\textnaira}{\HTMLUnicode{20A6}}
48 \renewcommand*{\textpeso}{\HTMLUnicode{20B1}}
49 \renewcommand*{\textrecipe}{\HTMLUnicode{211E}}
50 \renewcommand*{\textinterrobangdown}{\HTMLUnicode{2E18}}
51 \renewcommand*{\textperthousand}{\HTMLUnicode{2030}}
52 \renewcommand*{\textpertenthousand}{\HTMLUnicode{2031}}
53 \renewcommand*{\textbaht}{\HTMLUnicode{0E3F}}
54 \renewcommand*{\textdiscount}{\}%
55 \renewcommand*{\textservicemark}{\HTMLUnicode{2120}}
56 \else

```

§ 334.4.2 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.

```

57 \renewcommand*{\textdegree}{°}
58 \renewcommand*{\textcelsius}{°C}
59 \renewcommand*{\textohm}{Ω}
60 \renewcommand*{\textmu}{μ}
61 \renewcommand*{\textlquill}{{} }
62 \renewcommand*{\textrquill}{{} }
63 \renewcommand*{\textcircledP}{Ⓟ}
64 \renewcommand*{\texttwelveudash}{--}% emdash
65 \renewcommand*{\textthreequartersemdash}{--}% emdash
66 \renewcommand*{\textmho}{℧}
67 \renewcommand*{\textnaira}{₦}

```

```

68 \renewcommand*{\textpeso}{P}
69 \renewcommand*{\textrecipe}{R}
70 \renewcommand*{\textinterrobangdown}{ı}
71 \renewcommand*{\textperthousand}{‰}
72 \renewcommand*{\textpertenthousand}{‰.}
73 \renewcommand*{\textbaht}{฿}
74 \renewcommand*{\textdiscount}{\%}
75 \renewcommand*{\textservicemark}{®}
76 \fi

```

§ 334.5 HTML dicritics

For HTML, Unicode diacritical marks are used:

```

77 \renewcommand*{\capitalcedilla}[1]{\#1\HTMLUnicode{0327}}
78 \renewcommand*{\capitalogonek}[1]{\#1\HTMLUnicode{0328}}
79 \renewcommand*{\capitalgrave}[1]{\#1\HTMLUnicode{0300}}
80 \renewcommand*{\capitalacute}[1]{\#1\HTMLUnicode{0301}}
81 \renewcommand*{\capitalcircumflex}[1]{\#1\HTMLUnicode{0302}}
82 \renewcommand*{\capitaltilde}[1]{\#1\HTMLUnicode{0303}}
83 \renewcommand*{\capitaldieresis}[1]{\#1\HTMLUnicode{0308}}
84 \renewcommand*{\capitalhungarumlaut}[1]{\#1\HTMLUnicode{30B}}
85 \renewcommand*{\capitalring}[1]{\#1\HTMLUnicode{30A}}
86 \renewcommand*{\capitalcaron}[1]{\#1\HTMLUnicode{30C}}
87 \renewcommand*{\capitalbreve}[1]{\#1\HTMLUnicode{306}}
88 \renewcommand*{\capitalmacron}[1]{\#1\HTMLUnicode{304}}
89 \renewcommand*{\capitaldotaccent}[1]{\#1\HTMLUnicode{307}}

```

`\textcircled` becomes a span with a rounded border:

```

90 \renewcommand*{\textcircled}[1]{%
91 \InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{\#1}%
92 }
93 }% AtBeginDocument

```

§ 334.6 Inside a lateximage

When a `lateximage` is begun:

```

94 \appto\LWR@restoreorigformatting{%
95 \let\textdegree\LWR@origtextdegree%
96 \let\textcelsius\LWR@origtextcelsius%
97 \let\textohm\LWR@origtextohm%
98 \let\textmu\LWR@origtextmu%
99 \let\textlquill\LWR@origtextlquill%
100 \let\textrquill\LWR@origtextrquill%
101 \let\textcircledP\LWR@origtextcircledP%
102 \let\texttwelveudash\LWR@origtexttwelveudash%
103 \let\textthreequartersemdash\LWR@origtextthreequartersemdash%

```

```

104 \let\textmho\LWR@origtextmho%
105 \let\textnaira\LWR@origtextnaira%
106 \let\textpeso\LWR@origtextpeso%
107 \let\textrecipe\LWR@origtextrecipe%
108 \let\textinterrobangdown\LWR@origtextinterrobangdown%
109 \let\textperthousand\LWR@origtextperthousand%
110 \let\textpertenthousand\LWR@origtextpertenthousand%
111 \let\textbaht\LWR@origtextbaht%
112 \let\textdiscount\LWR@origtextdiscount%
113 \let\textservicemark\LWR@origtextservicemark%
114 \LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
115 \LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
116 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
117 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
118 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
119 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%
120 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
121 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
122 \LetLtxMacro\capitalring\LWR@origcapitalring%
123 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
124 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
125 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
126 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
127 \LetLtxMacro\textcircled\LWR@origtextcircled%
128 }

```

File 243 **lwarp-textfit.sty**

§ 335 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 244 **lwarp-textpos.sty**

§ 336 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 245 **lwarp-theorem.sty**

§ 337 Package **theorem**

(Emulates or patches code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by lwarp.

Table 14: Theorem package — CSS styling of theorems and proofs

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

§ 337.1 Remembering the theorem style

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \gdef\theoremstyle#1{%
4   \@ifundefined{th@#1}{\@warning
5     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
6     \theoremstyle{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% lwarp
8   }%
9   {%
10    \theoremstyle{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% lwarp
12  }%
13  \begingroup
14    \csname th@the\theoremstyle \endcsname
15  \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
16 \gdef\@xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
18   {%
19     \csedef{\LWR@thmstyle#1}{\LWR@newtheoremstyle}% 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 \theorem@style \expandafter
41             \endcsname \the \theorem@bodyfont \@thm{#1}{#2}}%
42         \global \expandafter \let \csname end#1\endcsname \@endtheorem
43         \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% 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 \theorem@style \expandafter
57                 \endcsname \the \theorem@bodyfont \@thm{#2}{#3}}%
58             \global \expandafter \let \csname end#1\endcsname \@endtheorem
59             \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% lwarp
60         }%
61     \fi}

```

§ 337.2 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 246 **lwarp-threeparttable.sty**

§ 338 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 {<text>}

Prints the table note item header inside a CSS class of tnoteitemheader.

2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}

Env **threeparttable** [*<alignment>*] To emulate threeparttable:

3 \newenvironment*{threeparttable}[1][b]{}{}

Env **tablenotes** [*<options>*]

```

4 \newenvironment*{tablenotes}[1] []
5 {%
6 \LWR@forcenewpage
7 \BlockClass{tnotes}%
8 \ltx@ifpackageloaded{enumitem}{%
9 \setlist[description]{format=\LWR@printtablenote}%
10 }{}%
11 \description%
12 }
13 {%
14 \enddescription%
15 \endBlockClass%
16 }
```

\tnote {<text>}


17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}

File 247 **lwarp-tikz.sty**

§ 339 Package **tikz**

(Emulates or patches code by TILL TANTAU.)

Pkg tikz **tikz** is supported.

 **displaymath and matrices** If using display math with `tikzpicture` or `\tikz`, along with matrices with the `&` character, the document must be modified as follows:

```
\usepackage{tikz}
\tikzset{every picture/.style={ampersand replacement=\&}}
```

and each instance of `&` in the **tikz** expression must be replaced with `\&`.

Accept all options for **lwarp-tikz**:

```
1 \LWR@ProvidesPackagePass{tikz}[2015/08/07]
```

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 **pgfpicture** The `\pgfpicture` environment is enclosed inside a `\lateximage`. Enclose the low-level `\pgfpicture` in a `lateximage`. This is also used by the higher-level `\tikz` and `tikzpicture`.

```
7 \preto\pgfpicture{%
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 \appto\endpgfpicture{\end{lateximage}}

```

Tikz is placed inside an SVG image, so use the original meanings of the following:

```

15 \LetLtxMacro\pgfutil@minipage\LWR@print@minipage
16 \let\pgfutil@endminipage\endLWR@print@minipage
17
18 \let\pgfutil@raggedleft\LWR@origraggedleft
19 \let\pgfutil@raggedright\LWR@origraggedright
20
21 \def\pgfutil@font@tiny{\LWR@origtiny}
22 \def\pgfutil@font@scriptsize{\LWR@origscriptsize}
23 \def\pgfutil@font@footnotesize{\LWR@origfootnotesize}
24 \def\pgfutil@font@small{\LWR@origsmall}
25 \def\pgfutil@font@normalsize{\LWR@orignormalsize}
26 \def\pgfutil@font@large{\LWR@origlarge}
27 \def\pgfutil@font@Large{\LWR@origLarge}
28 \def\pgfutil@font@huge{\LWR@orighuge}
29 \def\pgfutil@font@Huge{\LWR@origHuge}
30
31 \def\pgfutil@font@itshape{\LWR@origitshape}
32 \def\pgfutil@font@bfseries{\LWR@origbfseries}
33
34 \def\pgfutil@font@normalfont{\LWR@orignormalfont}

```

File 248 **lwarp-titlps.sty**

§ 340 Package **titlps**

(Emulates or patches code by JAVIER BEZOS.)

Pkg **titlps** **titlps** is loaded and used by **lwarp** during HTML output. All user options and macros are ignored and disabled.

Discard all options for **lwarp-titlps**:

for HTML output: 1 \LWR@ProvidesPackageDrop{titlps}

\pagestyle and \thispagestyle are already disabled in the **lwarp** code.

```

\newpagestyle    {\<name>} [<style>] {\<commands>}
                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}{}

\makefootrule
                  12 \newcommand*{\makefootrule}{}

\setmarkboth     {\langle code\rangle}
                  13 \newcommand{\setmarkboth}[1]{}

\widenhead
                  14 \NewDocumentCommand{\widenhead}{s o o m m}{}

```

```

\bottitlemarks
15 \newcommand*{\bottitlemarks}{}

\toptitlemarks
16 \newcommand*{\toptitlemarks}{}

\firsttitlemarks
17 \newcommand*{\firsttitlemarks}{}

\nexttitlemarks
18 \newcommand*{\nexttoptitlemarks}{}

\outertitlemarks
19 \newcommand*{\outertitlemarks}{}

\innertitlemarks
20 \newcommand*{\innertitlemarks}{}

\newtitlemark * {\langle name \rangle}
21 \NewDocumentCommand{\newtitlemark}{s m}{}

\pretitlemark * {\langle section \rangle} {\langle text \rangle}
22 \NewDocumentCommand{\pretitlemark}{s m m}{}

\ifsamemark {\langle group \rangle} {\langle command \rangle} {\langle true \rangle} {\langle false \rangle}
23 \newcommand{\ifsamemark}[4]{}

\setfloathead * [\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m}{}

\setfloatfoot * [\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m}{}

\nextfloathead * [\langle . \rangle] [\langle . \rangle] [\langle . \rangle] {\langle . \rangle} {\langle . \rangle} {\langle . \rangle} {\langle extra \rangle} [\langle which \rangle]

```

```

26 \NewDocumentCommand{\nextfloathead}{s o o m m m m m}{-}

\nextfloatfoot * [⟨.⟩] [⟨.⟩] [⟨.⟩] {⟨.⟩} {⟨.⟩} {⟨.⟩} {⟨extra⟩} [⟨which⟩]
27 \NewDocumentCommand{\nextfloatfoot}{s o o m m m m m}{-}

\nemarkset {⟨markset⟩}
28 \newcommand{\newmarkset}[1]{-}

\nextramark * {⟨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 249 **lwarp-titleref.sty**

§ 341 Package **titleref**

Pkg titleref **titleref** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{titleref}

```

2
3 \LetLtxMacro\title\ref\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 250 **lwarp-titlesec.sty**

§ 342 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}

```

\titlelabel {\langle label-format \rangle}
2 \newcommand*{\titlelabel}[1]{}
```

```

\titleformat* {\langle command \rangle} {\langle format \rangle}
```

```

\titleformat {\langle command \rangle} [\langle shape \rangle] {\langle format \rangle} {\langle label \rangle} {\langle sep \rangle} {\langle before \rangle} [\langle after \rangle]
3 \newcommand\titleformat{%
4   \@ifstar{\ttl@format@s}%
5   {\ttl@format@i}}
6 \newcommand{\ttl@format@s}[1]{}
7 \NewDocumentCommand{\ttl@format@i}{m o m m m o}{}
```

\chaptertitlename

```

8 \@ifundefined{@chapapp}{\let\@chapapp\chaptername}{}
9 \newcommand\chaptertitlename{\@chapapp}
```

```

\titlespacing * {\langle command \rangle} {\langle left \rangle} {\langle before \rangle} {\langle after \rangle} [\langle right \rangle]
```

```

10 \NewDocumentCommand{\titlespacing}{s m m m 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}

\titlename * [\langle align \rangle] {\langle material \rangle}
19 \NewDocumentCommand{\titlename}{s o m}{}

\titlerule [\langle height \rangle]
20 \providecommand*\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
21 \newcommand*{\ttl@rule}[1] [] {}
22 \newcommand*{\ttl@row}[2] [] {}

\iftitlemeasuring {\langle true \rangle} {\langle false \rangle}
23 \newcommand{\iftitlemeasuring}[2] {\#2}

\assignpagestyle {\langle command \rangle} {\langle pagestyle \rangle}

```

```
25 \NewDocumentCommand{\titleclass}{m o m o}
```

```
9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}
```

<code>\contentspage</code>	<code>[\format]</code>
	10 \newcommand{\contentspage}[1] [] {\thecontentspage}
<code>\contentspush</code>	<code>{\text}</code>
	11 \newcommand{\contentspush}[1] {}
<code>\contentsuse</code>	<code>{\name} {\text}</code>
	12 \newcommand{\contentsuse}[2] {}
<code>\startcontents</code>	<code>[\name]</code>
	13 \newcommand*{\startcontents}[1] [] {}
<code>\stopcontents</code>	<code>[\name]</code>
	14 \newcommand*{\stopcontents}[1] [] {}
<code>\resumecontents</code>	<code>[\name]</code>
	15 \newcommand*{\resumecontents}[1] [] {}
<code>\printcontents</code>	<code>[\name] {\prefix} {\start} {\code}</code>
	16 \newcommand{\printcontents}[4] [] {}
<code>\startlist</code>	<code>[\name] {\list}</code>
	17 \newcommand{\startlist}[2] [] {}
<code>\stoplist</code>	<code>[\name] {\list}</code>
	18 \newcommand{\stoplist}[2] [] {}
<code>\resumelist</code>	<code>[\name] {\list}</code>
	19 \newcommand{\resumelist}[2] [] {}
<code>\printlist</code>	<code>[\name] {\list} {\prefix} {\code}</code>
	20 \newcommand{\printlist}[4] [] {}

File 252 **lwarp-titling.sty**

§ 344 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 **authblk** and **titling**. If both are used, **authblk** should be loaded before **titling**.

 **load order**

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

The various **titling** footnote restyling commands have no effect.

Pass all options to **lwarp-titling**:

for HTML output: 1 \LWR@ProvidesPackagePass{titling}

\@bsmtitleempty Patch \@bsmtitleempty:

```
2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*{\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 }
```

\keepthetitle Patch \keepthetitle:

```
6 \let\LWR@origkeepthetitle\keepthetitle
7 \renewcommand*{\keepthetitle}{%
8 \LWR@orig@keepthetitle%
9 }
```

\killtitle Patch \killtitle:

```
10 \let\LWR@origkilltitle\killtitle
11 \renewcommand*{\killtitle}{%
12 \LWR@orig@killtitle%
13 }
```

Env **titlingpage**

```
14 \renewenvironment*{titlingpage}
15 {%
```

Start an HTML titlepage div:

```
16 \LWR@printpendingfootnotes
17 \begin{titlepage}
```

Prepare for a custom version of \maketitle inside the titlingpage:

```
18 \LWR@maketitlesetup
19 \let\maketitle\LWR@titlingmaketitle
20 }
21 {
```

At the end of the environment, end the HTML titlepage div:

```
22 \end{titlepage}
23 }
```

Patch the pre/post title/author/date to add HTML tags, then initilize:

```
24
25 \pretitle{}
26 \posttitle{}
27
28 \preauthor{}
29 \postauthor{}
30
31 \predate{}
32 \postdate{}
```

\LWR@maketitlesetup Patches \thanks macros.

```
33 \renewcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
34 \def\@makefnmark{\textsuperscript{\@thefnmark}}
```

```
\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}
```

Redefine the footnote text:

```
35 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
36 \makethanksmark~%

      \makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
                        \@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
37 ##1%
38 }% \@makefnmark
39 }
```

`\thanksfootmark`

```
40 \renewcommand{\thanksfootmark}{%
41   \hb@xt@\thanksmarkwidth{\hfil\normalfont%
42     \thanksscript{%
43       \thanksfootpre \tamark \thanksfootpost%
44     }%
45   }%
46 }
```

`\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.

```
47 \renewcommand*\maketitle{%
```

An HTML titlepage <div> is used for all classes.

```
48 \begin{titlepage}
```

Select which kind of footnote marks to use:

```
49 \@bsmarkseries
```

Set up special patches:

```
50 \LWR@maketitlesetup
```

Typeset the title, etc:

```
51 \@maketitle
```

Immediately generate any \thanks footnotes:

```
52 \@thanks
```

Close the HTML titlepage div:

```
53 \end{titlepage}
```

Reset the footnote counter:

```
54 \@bscontmark
55 }
```

`\@maketitle` Typesets the title, etc. Patched for HTML.

```
56 \DeclareDocumentCommand{\@maketitle}{-}{%
57   \maketitlehooka
58   {
59     \LWR@stoppars\LWR@htmltag{\LWR@tagtitle}
60     \@bsprettitle \@title \@bsposttitle
61     \LWR@htmltag{\LWR@tagtitleend}\LWR@startpars
62   }
63   \maketitlehookb
64   {
65     \begin{BlockClass}{author}
66     \renewcommand{\and}{
67       \end{BlockClass}
68       \begin{BlockClass}{oneauthor}
69     }
70     \begin{BlockClass}{oneauthor}
71     \@bspreadauthor \@author \@bspostauthor
72     \end{BlockClass}
73     \end{BlockClass}
74   }
75   \maketitlehookc
76   {
77     \begin{BlockClass}{titledate}
78     \@bspredate \@date \@bspostdate
79     \end{BlockClass}
80   }
81   \maketitlehookd
82 }
```

`\LWR@titlingmaketitle` \maketitle for use inside an HTML titlingpage environment.

```
83 \renewcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
84 \@thanks
```

Select which kind of footnote marks to use:

```
85 \@bsmarkseries
```

Set up special patches:

```
86 \LWR@maketitlesetup
```

Typeset the title, etc:

```
87 \@maketitle
```

Immediately generate any \@thanks footnotes:

```
88 \@thanks
```

Reset the footnote counter:

```
89 \@bscontmark
90 }
```

```
\thanksmarkseries {\langle series \rangle}
```

Sets the type of footnote marks used by \@thanks, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```
91 \renewcommand{\thanksmarkseries}[1]{%
92 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
93 }
```

Set default titlepage thanks footnote marks. See section [62.7](#).

```
94 \@ifclassloaded{memoir}{
95   \thanksmarkseries{arabic}
96 }{% not memoir
97 \if@titlepage
98   \thanksmarkseries{arabic}
99 \else
100  \thanksmarkseries{fnsymbol}
101 \fi
102 }% not memoir
```

File 253 **lwarp-tocbasic.sty**

§ 345 Package **tocbasic**

(Emulates or patches code by MARKUS KOHM.)

Pkg **tocbasic** **tocbasic** is patched for use by **lwarp**.

This package may be loaded standalone, but is also loaded automatically if **koma-script** classes are in use. `\DeclareDocumentCommand` is used to overwrite the **koma-script** definitions.

for HTML output:

```

1 \LWR@ProvidesPackagePass{tocbasic}

2 \DeclareDocumentCommand{\usetocbasicnumberline}{o}{-}{}
3 \DeclareDocumentCommand{\DeclareTOCStyleEntry}{o m m}{-}{}
4 \DeclareDocumentCommand{\DeclareTOCEntryStyle}{m o m}{-}{}
5 \DeclareDocumentCommand{\DefineTOCEntryOption}{m o m}{-}{}
6 \DeclareDocumentCommand{\DefineTOCEntryBooleanOption}{m o m m m}{-}{}
7 \DeclareDocumentCommand{\DefineTOCEntryCommandOption}{m o m m m}{-}{}
8 \DeclareDocumentCommand{\DefineTOCEntryIfOption}{m o m m m}{-}{}
9 \DeclareDocumentCommand{\DefineTOCEntryLengthOption}{m o m m m}{-}{}
10 \DeclareDocumentCommand{\DefineTOCEntryNumberOption}{m o m m m}{-}{}
11 \DeclareDocumentCommand{\CloneTOCEntryStyle}{m m}{-}{}
12 \DeclareDocumentCommand{\TOCEntryStyleInitCode}{m m}{-}{}
13 \DeclareDocumentCommand{\TOCEntryStyleStartInitCode}{m m}{-}{}

```

File 254 **lwarp-tocbibind.sty**

§ 346 Package **tocbibind**

(Emulates or patches code by PETER WILSON.)

Pkg **tocbibind** **tocbibind** is patched for use by **lwarp**.

Opt **xindyLanguage** The **lwarp** package option **xindyLanguage** sets the language used by **xindy**. This is passed to **xindy** using its `-L` option, and is used for both index and glossary generation.
 Default: **english**

Opt **xindyCodepage** The option **xindycodepage** sets the codepage used by **xindy**. This is passed to **xindy** using its `-C` option, and is used for both index and glossary generation.
 Default: **utf8**

Pkg **tocloft** If using **tocloft** with **tocbibind**, **anonchap**, **fncychap**, or other packages which change



tocloft & other packages

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.

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

Inline, with a manual TOC entry:

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

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

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

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

Inline, with an automatic TOC entry:

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

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

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

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

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

for HTML output:

```
1 \let\simplechapterdelim\relax
2
3 \LWR@ProvidesPackagePass{tocbibind}
```

```

4 \renewenvironment{theindex}%
5 {%
6     \if@bibchapter
7         \if@donumindex
8             \chapter{\indexname}
9         \else
10            \if@dotocind
11                \chapter*{\indexname}
12                \addcontentsline{toc}{chapter}{\indexname}
13            \else
14                \chapter*{\indexname}
15            \fi
16        \fi
17    \else
18        \if@donumindex
19            \section{\indexname}
20        \else
21            \if@dotocind
22                \section*{\indexname}
23                \addcontentsline{toc}{\@tocextra}{\indexname}
24            \else
25                \section*{\indexname}
26            \fi
27        \fi
28    \fi
29 \let\item\LWR@indexitem%
30 \let\subitem\LWR@indexsubitem%
31 \let\subsubitem\LWR@indexsubsubitem%
32 }{}

```

The following code is shared by **anonchap**.

```

33 \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\@seccntformat%
41 }

```

File 255 **lwarp-tocenter.sty**

§ 347 Package **tocenter**

Pkg tocenter **tocenter** is ignored.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{tocenter}

2 \NewDocumentCommand{\ToCenter}{s o m m}{\}
3 \NewDocumentCommand{\FromMargins}{s o m m m m}{\}
```

File 256 **lwarp-tocloft.sty**

§ 348 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

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

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}{\}
```

```
\cftlofttitlefont
9 \newcommand*{\cftlofttitlefont}{}

\cftafterlofttitle
10 \newcommand*{\cftafterlofttitle}{}

11 \newlength{\cftbeforelofttitleskip}
12 \newlength{\cftafterlofttitleskip}

\cftmarklot
13 \newcommand*{\cftmarklot}{}

\cftlottitlefont
14 \newcommand*{\cftlottitlefont}{}

\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    {\langle length \rangle}
23 \DeclareDocumentCommand{\cftsetrmarg}{m}{}
```

```
\cftpnumalign   {\langle alignment \rangle}
24 \DeclareDocumentCommand{\cftpnumalign}{m}{}

25 \LWR@providelength{\cftparskip}
```

The part-related items are also provided by **memoir**:

```
26 \LWR@providelength{\cftbeforepartskip}
27 \LWR@providelength{\cftpartinindent}
28 \LWR@providelength{\cftpartinnumwidth}
29 \providecommand*\cftpartinfont{}
30 \providecommand*\cftpartinpresnum{}
31 \providecommand*\cftpartinftersnum{}
32 \providecommand*\cftpartinftersnumb{}
33 \providecommand*\cftpartinleader{}
34 \providecommand*\cftpartinldotsep{1}
35 \providecommand*\cftpartinpagefont{}
36 \providecommand*\cftpartinfterpnum{}

```

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*\cftchapftersnum{}
43 \newcommand*\cftchapftersnumb{}
44 \newcommand*\cftchapleader{}
45 \newcommand*\cftchapldotsep{1}
46 \newcommand*\cftchappagefont{}
47 \newcommand*\cftchapfterpnum{}

```

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{l@#2}##1##2{%
168     \hypertocfloat{1}{#2}{#3}{##1}{##2}%
169     \def\cftwhatismynum{#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}
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 257 **lwarp-tocstyle.sty**

§ 349 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 258 **lwarp-todo.sty**

§ 350 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 259 **lwarp-todonotes.sty**

§ 351 Package **todonotes**

(Emulates or patches code by HENRIK SKOV MIDTIBY.)

Pkg **todonotes** **todonotes** is emulated.

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

for HTML output:

```

1 \LWR@ProvidesPackagePass{todonotes}

2 \if@todonotes@disabled
3 \else
4
5 \newcommand{\ext@todo}{tdo}
6
7 \renewcommand{\l@todo}[2]{\hypertocfloat{1}{todo}{ldo}{#1}{#2}}

8 \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 260 **lwarp-transparent.sty**

§ 352 Package **transparent**

(Emulates or patches code by HEIKO OBERDIEK.)

Pkg transparent Emulated. `\texttransparent` works for inline objects. `\transparent` only works for `\includegraphics`.

 **Not \LaTeX !** Note that **transparent** does not work with \LaTeX .

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 261 **lwarp-trimclip.sty**

§ 353 Package **trimclip**

Pkg trimclip **trimclip** is nullified.

for HTML output: 1 \LWR@ProvidesPackageDrop{trimclip}

The third argument, the text, is not touched. This allows `\bgroup / \egroup`, and verbatim content.

```

2 \csdef{trimbox}{\@ifstar\@gobble\@gobble}
3 \csletcs{trimbox*}{trimbox}
4 \def\endtrimbox{}
5 \csletcs{endtrimbox*}{endtrimbox}

```

```

6
7 \csletcs{clipbox}{trimbox}
8 \csletcs{clipbox*}{trimbox}
9 \csletcs{endclipbox}{endtrimbox}
10 \csletcs{endclipbox*}{endtrimbox}
11
12 \csletcs{marginbox}{trimbox}
13 \csletcs{marginbox*}{trimbox}
14 \csletcs{endmarginbox}{endtrimbox}
15 \csletcs{endmarginbox*}{endtrimbox}

```

File 262 **lwarp-trivfloat.sty**

§ 354 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}

```

§ 354.1 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 263 `lwarp-turnthepage.sty`

§ 355 Package **turnthepage**

Pkg `turnthepage` **turnthepage** is ignored.

for HTML output: 1 `\LWR@ProvidesPackageDrop{turnthepage}`

2 `\newcommand{\turnthepage}{}{}`

File 264 `lwarp-typearea.sty`

§ 356 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 265 **lwarp-ulem.sty**

\$ 357 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: auto}%
7 {uline}{#1}%
8 }
9
10 \NewDocumentCommand{\uuline}{+m}{%
11 \LWR@HTMLtextstyle%
12 {%
13 text-decoration:underline; text-decoration-skip: auto;%

```

```

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: auto;%
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: auto;%
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: auto;%
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}{\}
74 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}
75
76 \DeclareOption*{}
77 \ProcessOptions\relax% original LaTeX code
```

File 266 **lwarp-underscore.sty**

\$ 358 Package **underscore**

Pkg underscore **underscore** is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{underscore}

File 267 **lwarp-upref.sty**

\$ 359 Package **upref**

Pkg upref Ignored.

for HTML output: Discard all options for **lwarp-upref**:

```
1 \LWR@ProvidesPackageDrop{upref}
```

File 268 **lwarp-url.sty**

\$ 360 Package **url**

(Emulates or patches code by DONALD ARSENEAU.)

Pkg url **url** is patched for use by **lwarp**.

for HTML output: `1 \LWR@ProvidesPackagePass{url}`

url uses math mode to print its string inside a group, so the original meaning of math is restored first.

```
2 \LetLtxMacro\LWR@url@origUrl@FormatString\Url@FormatString
3
4 \renewcommand*{\Url@FormatString}{%
5   \InlineClass{verbatim}{%
6     \LWR@restoreorigformatting%
7     \LWR@url@origUrl@FormatString%
8   }%
9 }
```

File 269 **lwarp-verse.sty**

\$ 361 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}
```

When using **verse** or **memoir**, always place a `\\` after each line.

`\attrib` The documentation for the **verse** and **memoir** packages suggest defining an `\attrib` command, which may already exist in current documents, but it will only work for print output. **lwarp** provides `\attribution`, which works for both print and HTML


output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

```
\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}
```

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

 **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%
```

Pkg verse Cls memoir <code>\flagverse</code> Len <code>\vleftskip</code>	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:
---	---

```
7 \ifdef{\vleftskip}{%
8 \setlength{\vleftskip}{\HTMLvleftskip}
9 \setlength{\leftmargini}{\HTMLleftmargini}
10 }{}
```

```

11 \LWR@forcenewpage
12 \LWR@atbeginverbatim{3}{verse}%
13 }

```

After the end of the verse environment, which places the <pre> tag at the regular left margin:

```

14 \AtEndEnvironment{verse}{%
15 \leavevmode%
16 \LWR@afterendverbatim{1}%
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 270 **lwarp-vertbars.sty**

§ 362 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 271 **lwarp-vmargin.sty**

§ 363 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 272 **lwarp-vowel.sty**

§ 364 Package **vowel**

(Emulates or patches code by FUKUI REI.)

Pkg vowel **vowel** is patched for use by **lwarp**.

This package has been tested with **pdflatex** and the Type 1 TIPA fonts using the following package load sequence:

```

\usepackage[T3,T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[noenc]{tipa}
\usepackage{vowel}

```

for HTML output:

```

1 \LWR@ProvidesPackagePass{vowel}

```

```

2 \renewenvironment{vowel}[1] []
3   {%
4     \begin{lateximage}[(-vowel--\packagediagramname)]%
5     \@vowel[#1]%
6   }
7   {%
8     \@@vowel%
9     \end{lateximage}%
10  }

```

File 273 **lwarp-vwcol.sty**

\$ 365 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 {\<style tag>}

```

2 \newcommand*{\LWR@vwcol@addrule}[1]{%
3   \appto{\LWR@vwcolstyle}{%
4     #1: %
5     \LWR@printlength{\vwcol@rule} solid \LWR@origpound\LWR@vwcol@rulecolor ; %
6   }%
7 }

```

\LWR@vwcol@addrule {\<style tag>}

```

8 \newcommand*{\LWR@vwcol@addgap}[1]{%
9   \appto{\LWR@vwcolstyle}{%
10    #1: %
11    \LWR@printlength{\vwcol@sep} ; %
12  }%
13 }

```

Env vwcol {\<key/values>}

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 }

```

File 274 **lwarp-wallpaper.sty**

§ 366 Package **wallpaper**

(Emulates or patches code by MICHAEL H.F. WILKINSON.)

Pkg wallpaper **wallpaper** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{wallpaper}

```

2 \newcommand*{\CenterWallPaper}[2]{}
3 \newcommand*{\ThisCenterWallPaper}[2]{}
4 \newcommand*{\TileWallPaper}[3]{}
5 \newcommand*{\ThisTileWallPaper}[3]{}
6 \newcommand*{\TileSquareWallPaper}[2]{}
7 \newcommand*{\ThisTileSquareWallPaper}[2]{}
8 \newcommand*{\ULCornerWallPaper}[2]{}
9 \newcommand*{\ThisULCornerWallPaper}[2]{}
10 \newcommand*{\LLCornerWallPaper}[2]{}
11 \newcommand*{\ThisLLCornerWallPaper}[2]{}
12 \newcommand*{\URCornerWallPaper}[2]{}
13 \newcommand*{\ThisURCornerWallPaper}[2]{}

```

```

14 \newcommand*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}
16 \newcommand*{\ClearWallPaper}{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}

```

File 275 **lwarp-wasysym.sty**

§ 367 Package **wasysym**

Pkg wasysym **wasysym** does not work with **pdftotext**.

for HTML output: 1 \LWR@loadnever{wasysym}{textcomp, amssymb, amsfonts, mnsymbol, fdsymbol}

File 276 **lwarp-watermark.sty**

§ 368 Package **watermark**

(Emulates or patches code by ALEXANDER I. ROZHENKO.)

Pkg watermark **watermark** is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{watermark}

```

2 \newcommand{\watermark}[1]{}
3 \newcommand{\leftwatermark}[1]{}
4 \newcommand{\rightwatermark}[1]{}
5 \newcommand{\thiswatermark}[1]{}
6 \newcommand{\thispageheading}[1]{}

```

File 277 **lwarp-wrapfig.sty**

§ 369 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 278 **lwarp-xcolor.sty**

§ 370 Package **xcolor**

(Emulates or patches code by DR. UWE KERN.)

Pkg xcolor **xcolor** is supported by lwarp.

§ 370.1 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.

colored tables \rowcolors is supported, except that the optional argument is ignored so far.

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

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

§ 370.2 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 81. If **xcolor** has been loaded, the print-mode `\fcolorbox` is modified to accept a background color of none, and additional definitions are created for **lwarp**'s new macros print-mode macros `\colorboxBlock`, `\fcolorboxBlock`, and `fcolorminipage`. The HTML versions of these macros will already have been created by **lwarp-xcolor** if it has been loaded.

For use inside an HTML `lateximage`, `\LWR@restoreorigformatting` is appended to temporarily set these functions to their print-mode versions.

In a lateximage in HTML mode: `\LWR@restoreorigformatting` temporarily restores the print-mode definitions of **xcolor**'s functions. See `\LWR@restoreorigformatting` on page 456.

\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 863.

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 863.

HTML lateximage: Remembers and reuses the print version.

\colorboxBlock:

Print: Becomes `\colorbox`.

HTML: Newly defined by **lwarp-xcolor** to use a `<div>`, page 864.

HTML lateximage: Remembers and reuses the print version `\colorbox`.

\fcolorbox:

Print: Modified to allow a background of none.

`\LWR@print@fcolorbox` at section 81

HTML: Redefined by **lwarp-xcolor**, page 864.

HTML lateximage: Remembers and reuses the print version.

\fcolorboxBlock:

Print: Becomes `\fcolorbox`. Section 81

HTML: Newly defined by **lwarp-xcolor** to use a `<div>`, page 865.

HTML lateximage: Remembers and reuses the print version `\fcolorbox`.

fcolorminipage:

Print: Newly defined in the **lwarp** core.

`LWR@print@fcolorminipage` at section 81

HTML: Newly defined by **lwarp-xcolor**, page 866.

HTML lateximage: Uses the print version.

\boxframe:

Print: Used as-is.

HTML: Redefined by **lwarp-xcolor**, page 866.

HTML lateximage: Remembers and reuses the print version.

§ 370.3 Package loading

for HTML output:

```
1 \LWR@ProvidesPackagePass{xcolor}

2 \begin{warpHTML}
```

§ 370.4 Remembering and restoring original definitions

Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```
3 \LetLtxMacro\LWR@print@pagecolor\pagecolor
4 \LetLtxMacro\LWR@print@nopagecolor\nopagecolor
```

`\LWR@restoreorigformatting` Inside a `lateximage` the following gets restored to their print-mode actions:

```
5 \appto\LWR@restoreorigformatting{%
6 \LetLtxMacro\pagecolor\LWR@print@pagecolor%
7 \LetLtxMacro\nopagecolor\LWR@print@nopagecolor%
8 }
```

§ 370.5 HTML color style

Sets `\LWR@tempcolor` to the current color.

```
\LWR@findcurrenttextcolor
9 \renewcommand*{\LWR@findcurrenttextcolor}{%
10 \protect\colorlet{\LWR@current@color}{.}%
11 \protect\convertcolorspec{named}{\LWR@current@color}{HTML}\LWR@tempcolor%
12 }
```

Prints a color style for the current color.

```
\LWR@currenttextcolorstyle
13 \newcommand*{\LWR@currenttextcolorstyle}{%
14 \LWR@findcurrenttextcolor%
15 \ifdefstring{\LWR@tempcolor}{000000}%
16 {}%
17 {color: \LWR@origpound\LWR@tempcolor ; }%
18 }
```

`\LWR@textcurrentcolor` `{<text>}` Like `\textcolor` but uses the current `\color` instead.

```
19 \DeclareDocumentCommand{\LWR@textcurrentcolor}{m}{%
20 \begingroup%
21 \LWR@FBcancel%
22 \LWR@findcurrenttextcolor%
23 \InlineClass[color:\LWR@origpound\LWR@tempcolor]{textcolor}{%
24   \renewcommand*{\LWR@currenttextcolor}{\LWR@origpound\LWR@tempcolor}%
25   #1%
```

```

26 }%
27 \endgroup%
28 }

```

`\LWR@colorstyle` $\{\langle 2: model \rangle\} \{\langle 3: color \rangle\}$

For a color style, prints the color converted to HTML colors.

```

29 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
30 \begingroup%
31 \LWR@FBcancel%

```

Use the **xcolor** package to convert to an HTML color space:

```

32 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%

```

Print the converted color:

```

33 \LWR@origpound\LWR@tempcolor%
34 \endgroup%
35 }

```

`\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.

```

36 \NewDocumentCommand{\LWR@backgroundcolor}{0{named} m m}{%
37 \begingroup%
38 \LWR@FBcancel%
39 \InlineClass[background:\LWR@colorstyle{#1}{#2}]{backgroundcolor}{%
40 #3%
41 }%
42 \endgroup%
43 }

```

§ 370.6 HTML border

`\LWR@borderpadding` $\{\langle colorstyle \rangle\} \{\langle color \rangle\}$ Prints the HTML attributes for a black border and padding.

`\LWR@forceminwidth` must be used first in order to set the border width.

```

44 \newcommand*{\LWR@borderpadding}[2]{%
45 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
46 padding:\LWR@printlength{\fboxsep}%
47 }

```

§ 370.7 High-level macros

`\textcolor` [*model*] {*color*} {*text*}

Converted into an HTML hex color span.

```
48 \NewDocumentCommand{\LWR@HTML@textcolor}{o m m}{%
49 \begingroup%
```

Set the PDF color, to be picked up by SVG math if possible.

The print-mode `\color` command cannot accept the named option with color mixing, but it works with no option at all.

```
50 \IfValueTF{#1}{%
51   \color[#1]{#2}%
52 }{%
53   \color{#2}%
54 }%

55 \LWR@FBcancel%
56 \IfValueTF{#1}{%
57 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
58 \renewcommand*\LWR@currenttextcolor{\LWR@origpound\LWR@tempcolor}%
59 #3%
60 }%
61 }{%
62 \InlineClass[color:\LWR@colorstyle{named}{#2}]{textcolor}{%
63 \renewcommand*\LWR@currenttextcolor{\LWR@origpound\LWR@tempcolor}%
64 #3%
65 }%
66 }%
67 \endgroup%
68 }
69
70 \LWR@formatted{textcolor}
```

`\pagecolor` [*model*] {*color*}

Ignored. Use CSS instead.

```
71 \renewcommand*\pagecolor}[2] [named] {}
```

`\nopagecolor` Ignored.

```
72 \renewcommand*\nopagecolor[] {}
```

`\colorbox` [*model*] {*color*} {*text*}

Converted into an HTML hex background color ``.

```

73 \NewDocumentCommand{\LWR@HTML@colorbox}{O{named} m +m}{%
74 \begingroup%
75 \LWR@FBcancel%
76 \InlineClass[%
77 background:\LWR@colorstyle{#1}{#2} ; %
78 padding:\LWR@printlength{\fboxsep}%
79 ]{colorbox}{#3}%
80 \endgroup%
81 }
82
83 \AtBeginDocument{
84 \LWR@formatted{colorbox}
85 }

```

`\colorboxBlock` [*model*] {*color*} {*text*}

Converted into an HTML hex background color `<div>`.

```

86 \NewDocumentCommand{\LWR@HTML@colorboxBlock}{O{named} m +m}{%
87 \begingroup%
88 \LWR@FBcancel%
89 \begin{BlockClass}[%
90 background:\LWR@colorstyle{#1}{#2} ; %
91 padding:\LWR@printlength{\fboxsep}%
92 ]{colorboxBlock}
93 #3
94 \end{BlockClass}%
95 \endgroup%
96 }
97
98 \AtBeginDocument{
99 \LWR@formatted{colorboxBlock}
100 }

```

`\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.

```

101 \NewDocumentCommand{\LWR@HTML@fcolorbox}{O{named} m O{named} m +m}{%
102 \LWR@traceinfo{HTML fcolorbox #2 #4}%

```

```

103 \begingroup%
104 \LWR@FBcancel%
105 \LWR@forceminwidth{\fboxrule}%
106 \ifthenelse{\equal{#4}{none}}%
107 {% no background color
108   \InlineClass[%
109     \LWR@borderpadding{#1}{#2}%
110   ]{\fcolorbox}{#5}%
111 }%
112 {% yes background color
113   \InlineClass[%
114     \LWR@borderpadding{#1}{#2} ; %
115     background:\LWR@colorstyle{#3}{#4}%
116   ]{\fcolorbox}{#5}%
117 }%
118 \endgroup%
119 }
120
121 \AtBeginDocument{
122 \LWR@formatted{\fcolorbox}
123 }

```

`\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.

```

124 \NewDocumentCommand{\LWR@HTML@fcolorboxBlock}{0{named} m O{named} m +m}{%
125 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
126 \begingroup%
127 \LWR@FBcancel%
128 \LWR@forceminwidth{\fboxrule}%
129 \ifthenelse{\equal{#4}{none}}%
130 {% no background color
131   \begin{BlockClass}[%
132     \LWR@borderpadding{#1}{#2}%
133   ]{\fcolorboxBlock}
134   #5
135   \end{BlockClass}%
136 }%
137 {% yes background color
138   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
139   \begin{BlockClass}[%
140     background:\LWR@origpound\LWR@tempcolortwo\ ; %
141     \LWR@borderpadding{#1}{#2}%
142   ]{\fcolorboxBlock}
143   #5

```

```

144 \end{BlockClass}%
145 }%
146 \endgroup%
147 \LWR@traceinfo{HTML fcolorboxBlock done}%
148 }
149
150 \AtBeginDocument{
151 \LWR@formatted{fcolorboxBlock}
152 }

```

Creates a framed HTML <div> around its contents.

A print-output version is defined in the **lwarp** core: section 81

```

\LWR@subfcolorminipage {<framemodel>} {<framecolor>} {<background tag>} {<height>}

153 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
154 \begin{BlockClass}[%
155 #3%
156 \LWR@borderpadding{#1}{#2} ; %
157 \IfValueT{#4}{height:\LWR@printlength{\LWR@tempheight} ; }%
158 width:\LWR@printlength{\LWR@tempwidth}%
159 ]{fcolorminipage}%
160 }

Env fcolorminipage [{1:framemodel}] {<2:framecolor>} [{3:boxmodel}] {<4:boxcolor>} [{5:align}] [{6:height}]
[<7:inner-align>] {<8:width>}

161 \NewDocumentEnvironment{LWR@HTML@fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
162 {%
163 \LWR@FBcancel%
164 \setlength{\LWR@tempwidth}{#8}%
165 \IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
166 \LWR@forceminwidth{fboxrule}%
167 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
168 \ifthenelse{\equal{#4}{none}}%
169 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
170 {%
171 \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
172 \LWR@subfcolorminipage{#1}{#2}{background:\LWR@origpound\LWR@tempcolortwo\ ; }{#6}%
173 }%
174 }
175 {\end{BlockClass}}
176
177 \AtBeginDocument{
178 \LWR@formattedenv{fcolorminipage}
179 }

```

`\boxframe` $\{\langle width \rangle\} \{\langle height \rangle\} \{\langle depth \rangle\}$

The depth is added to the height, but the box is not decended below by the depth.
`\textcolor` is honored.

```

180 \newcommand*{\LWR@HTML@boxframe}[3]{%
181 {%
182 \setlength{\LWR@tempwidth}{#1}%
183 \setlength{\LWR@tempheight}{#2}%
184 \addtolength{\LWR@tempheight}{#3}%
185 \LWR@forceminwidth{\fboxrule}%
186 \LWR@findcurrenttextcolor%
187 \InlineClass[%
188 display:inline-block ; %
189 border:\LWR@printlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
190 width:\LWR@printlength{\LWR@tempwidth} ; %
191 height:\LWR@printlength{\LWR@tempheight}%
192 ]{\boxframe}{}%
193 }%
194 }
195
196 \LWR@formatted{boxframe}

```

§ 370.8 Row colors

`\rowc@l@rs` $[\langle cmds \rangle] \{\langle startrow \rangle\} \{\langle odd color \rangle\} \{\langle even color \rangle\}$

```

197 \newcommand*{\LWR@xcolortempcolor}{}
198
199 \def\rowc@l@rs[#1]#2#3#4%
200 {
201 \global\rownum=1
202 \global\@rowcolorstrue
203 \@ifxempty{#3}%
204 {\def\@oddrowcolor{\@norowcolor}}%
205 {%
206 \convertcolorspec{named}{#3}{HTML}\LWR@xcolortempcolor%
207 \edef\@oddrowcolor{%
208 \csdef{\LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
209 }%
210 }%
211 \@ifxempty{#4}%
212 {\def\@evenrowcolor{\@norowcolor}}%
213 {%
214 \convertcolorspec{named}{#4}{HTML}\LWR@xcolortempcolor%
215 \edef\@evenrowcolor{%
216 \csdef{\LWR@xcolorrowHTMLcolor}{\LWR@xcolortempcolor}%
217 }%
218 }%
219 \if@rowcmd

```

```

220 \def\@rowcolors
221 {%
222 % #1%
223 \if@rowcolors
224 % \noalign{%
225 % \relax\ifnum\rownum<#2\@norowcolor\else
226 % \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi\fi%
227 %}%
228 %\fi%
229}%
230 \else
231 \def\@rowcolors
232 {%
233 \if@rowcolors
234 % \ifnum\rownum<#2%
235 % \noalign{%
236 % \@norowcolor
237 %}%
238 % \else
239 % #1%
240 % \noalign{%
241 % \ifodd\rownum\@oddrowcolor\else\@evenrowcolor\fi%
242 %}%
243 %\fi
244 %\fi%
245}%
246 \fi
247 \ignorespaces%
248 }

```

`\@norowcolor` Turns off color for this row.

```

249 \def\@norowcolor{%
250 \renewcommand{\LWR@xcolorrowHTMLcolor}{}}%
251 }

```

`\@rowc@lors` Executed at the end of each row.

```

252 \def\@rowc@lors{%
253 % \noalign{%
254 % \global\advance\rownum\@ne%
255 %}%
256 %\@rowcolors%
257 }

```

```

258 \end{warpHTML}

```

File 279 **lwarp-xellipsis.sty**

§ 371 Package **xellipsis**

(Emulates or patches code by DONALD P. GOODMAN III.)

Pkg xellipsis **xellipsis** is patched for use by **lwarp**.

When non-zero, each of the spaces is converted to an HTML thin unbreakable space.

for HTML output:

```

1 \LWR@ProvidesPackagePass{xellipsis}

2 \newcommand*{\LWR@xellipsespace}[1]{%
3   \ifdim#1=0pt\else%
4     \ifdim#1<\fontdimen2\font%
5       \,%
6     \else%
7       ~%
8     \fi%
9 \fi%
10 }
11
12 \def\xelip{%
13   \mbox{%
14     \LWR@xellipsespace{\xelipprebef}%
15     \xelipprechar%
16     \LWR@xellipsespace{\xelippreaft}%
17     \LWR@xellipsespace{\xelipbef}%
18     \xelipchar%
19     \xel@loopi = 1%
20     \loop\ifnum\xelipnum>\xel@loopi%
21       \advance\xel@loopi by1%
22       \LWR@xellipsespace{\xelipgap}%
23       \xelipchar%
24     \repeat%
25     \LWR@xellipsespace{\xelipaft}%
26     \LWR@xellipsespace{\xelippostbef}%
27     \xelippostchar%
28     \LWR@xellipsespace{\xelippostaft}%
29 }%
30 }%
```


File 280 **lwarp-xfrac.sty**

§ 372 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 }

```

[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.

```

16 \DeclareInstance{xfrac}{default}{text}{
17 numerator-format = {%
18 \begingroup%
19 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
20 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
21 \endgroup%
22 },
23 denominator-format = {%
24 \begingroup%
25 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
26 \LWR@htmlsmallfontstart{ }\,#1\LWR@htmlsmallfontend%
27 \endgroup%
28 },

```

For **pdftotext**, do not scale the text:

```

29 scaling = false
30 }
31
32 \DeclareInstance{xfrac}{lmr}{text}{
33 numerator-format = {%
34 \begingroup%
35 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
36 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend%
37 \endgroup%
38 },
39 denominator-format = {%
40 \begingroup%
41 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
42 \LWR@htmlsmallfontstart{ }\,#1\LWR@htmlsmallfontend%
43 \endgroup%
44 },

```

For **pdftotext**, do not scale the text:

```

45 scaling = false

```

```

46 }
47
48 \DeclareInstance{xfrac}{lmss}{text}{
49 numerator-format = {%
50 \begingroup%
51 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
52 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
53 \endgroup%
54 },
55 denominator-format = {%
56 \begingroup%
57 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
58 \LWR@htmlsmallfontstart{\, #1 \LWR@htmlsmallfontend%
59 \endgroup%
60 },

```

For **pdftotext**, do not scale the text:

```

61 scaling = false
62 }
63
64 \DeclareInstance{xfrac}{lmtt}{text}{
65 numerator-format = {%
66 \begingroup%
67 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
68 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend%
69 \endgroup%
70 },
71 denominator-format = {%
72 \begingroup%
73 \RenewDocumentCommand{\scalebox}{m o m}{##3}%
74 \LWR@htmlsmallfontstart{\, #1 \LWR@htmlsmallfontend%
75 \endgroup%
76 },

```

For **pdftotext**, do not scale the text:

```

77 scaling = false
78 }

```

```

79 \end{warpHTML}

```

File 281 **lwarp-xltabular.sty**

§ 373 Package **xltabular**

(Emulates or patches code by ROLF NIEPRASCHK, HERBERT VOSS.)

Pkg xltabular **xltabular** is emulated by **lwarp**.

for HTML output: Relies on **tabularx**.

⚠ table numbering At present, an xltabular without a caption or with only a `\caption*` may be mis-numbered in HTML, so it may be necessary to place at the end of the table:

```
\warpHTMLonly{\addtocounter{table}{-1}}
```

```
1 \RequirePackage{tabularx}
2
3 \LWR@ProvidesPackageDrop{xltabular}
4
5 \DeclareDocumentEnvironment{xltabular}{o m m}
6 {\longtable{#3}}
7 {\endlongtable}
```

File 282 **lwarp-xltextra.sty**

§ 374 Package **xltextra**

(Emulates or patches code by WILL ROBERTSON, JONATHAN KEW.)

Pkg xltextra **xltextra** is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{xltextra}

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 283 **lwarp-xmpincl.sty**

§ 375 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 284 **lwarp-xpiano.sty**

§ 376 Package **xpiano**

(Emulates or patches code by ENRICO GREGORIO.)

Pkg xpiano **xpiano** is patched for use by **lwarp**.

for HTML output: 1 \LWR@ProvidesPackagePass{xpiano}

```

2 \ExplSyntaxOn
3 \NewDocumentCommand{\LWR@print@keyboard}{ O{}m }
4 {
5 \xpiano_keyboard:nn { #1 } { #2 }
6 }
7
8 \NewDocumentCommand{\LWR@HTML@keyboard}{ O{}m }
9 {
10 \begin{lateximage}*
11   [-xpiano--\packagegediagramname{}: \detokenize\expandafter{#2}]
```

```

12    [\detokenize\expandafter{#1}]
13 \xpiano_keyboard:nn { #1 } { #2 }
14 \end{lateximage}
15 }
16 \ExplSyntaxOff
17
18 \LWR@formatted{keyboard}

```

File 285 **lwarp-xtab.sty**

§ 377 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
...
\StopDefiningTabulars

```

See section 9.9.

⚠ 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 286 **lwarp-xurl.sty**

§ 378 Package **xurl**

Pkg xurl **xurl** is ignored.


for HTML output: 1 \LWR@ProvidesPackageDrop{xurl}

File 287 **lwarp-xy.sty**

§ 379 Package **xy**

(Emulates or patches code by KRISTOFFER H. ROSE, ROSS MOORE.)

Pkg xy **xy** is patched for use by **lwarp**.

 \xypolygon must be used inside the xy environment, or inside \xy ... \endxy.

for HTML output: 1 \LWR@ProvidesPackagePass{xy}

```

2 \AtBeginDocument{
3
4 \pretol{\xy}{\begin{lateximage}[(-xy--\packagedialogname)]}
5 \apptol{\endxy}{\end{lateximage}}
6
7 \@ifundefined{xymatrix}{}{
8 \LetLtxMacro\LWR@origxymatrix\xymatrix
9
10 \renewcommand{\xymatrix}[1]{%
11 \begin{lateximage}[(-xy- xymatrix \packagedialogname)]
12 \LWR@origxymatrix{#1}
13 \end{lateximage}
14 }
15 }
16
17 \@ifundefined{xygraph}{}{
18 \LetLtxMacro\LWR@origxygraph\xygraph
19
20 \renewcommand{\xygraph}[1]{%
21 \begin{lateximage}[(-xy- xygraph \packagedialogname)]
22 \LWR@origxygraph{#1}
23 \end{lateximage}

```

```

24 }
25 }
26
27 }

```

File 288 **lwarp-zwpagelayout.sty**

§ 380 Package **zwpagelayout**

(Emulates or patches code by ZDENĚK WAGNER.)

Pkg zwpagelayout **zwpagelayout** is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{zwpagelayout}

2 \def\noBboxes{}
3 \@onlypreamble\noBboxes
4
5 \expandafter\ifx\csname definecolor\endcsname\relax \else
6   \definecolor{cmykblack}{cmyk}{0,0,0,1}
7   \definecolor{grblack}{gray}{0}
8%   \ifzwpl@redefineblack
9%     \definecolor{black}{cmyk}{0,0,0,1}\color{black}
10%  \fi
11  \definecolor{cmykred}{cmyk}{0,1,1,0}
12  \definecolor{cmykgreen}{cmyk}{1,0,1,0}
13  \definecolor{cmykblue}{cmyk}{1,1,0,0}
14  \definecolor{rgbred}{rgb}{1,0,0}
15  \definecolor{rgbgreen}{rgb}{0,1,0}
16  \definecolor{rgbblue}{rgb}{0,0,1}
17%  \ifzwpl@redefinetcmyk
18%    \definecolor{red}{cmyk}{0,1,1,0}
19%    \definecolor{green}{cmyk}{1,0,1,0}
20%    \definecolor{blue}{cmyk}{1,1,0,0}
21%  \fi
22 \fi
23
24 \let\OverprintXeTeXExtGState\relax
25
26 \DeclareRobustCommand\SetOverprint{\ignorespaces}
27 \DeclareRobustCommand\SetKnockout{\ignorespaces}
28 \DeclareRobustCommand\textoverprint[1]{\SetOverprint#1}
29 \DeclareRobustCommand\textknockout[1]{\SetKnockout#1}
30
31 \def\SetPDFminorversion#1{}
32 \@onlypreamble\SetPDFminorversion
33

```

```

34 \newcommand*\Vcorr{}
35
36 \DeclareRobustCommand\vb[1] [] {}
37 \NewDocumentCommand{\NewOddPage}{* o}{}
38 \NewDocumentCommand{\NewEvenPage}{* o}{}
39 \def\SetOddPageMessage#{\gdef\ZW@oddwarning}
40 \def\SetEvenPageMessage#{\gdef\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{\LWR@origpound\,\arabic{page}}
56 \ifXeTeX
57 \def\ifcaseZWdriver{\ifcase2}
58 \else
59 \def\ifcaseZWdriver{\ifcase1}
60 \fi
61 \DeclareRobustCommand\ZWifdriver[2]{}

```

File 289 **lwarp-patch-komascript.sty**

§ 381 Package **patch-komascript**

Pkg lwarp-patch-komascript Patches for **komascript** classes.

lwarp loads this package when **scrbook**, **scrartcl**, or **scrreprt** classes are detected.

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

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

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

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

for HTML output: `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 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 290 **lwarp-patch-memoir.sty**


§ 382 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 382.1). 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}`

§ 382.1 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}% emulated below
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

```

§ 382.2 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@m{\@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}{*}

```

§ 382.3 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*{\stocksmallroyalvo}{}
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*{\stocksmalldemyvo}{}
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*{\pagesmalldemyvo}{}
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] {}

```

§ 382.4 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}{

```

§ 382.5 Titles

```

218 \csletcs{titlingpage*}{titlingpage}
219 \csletcs{endtitlingpage*}{endtitlingpage}
220 \let\titlingpageend\relax
221 \newcommand{\titlingpageend}[2]{
222 \let\andnext\and
223 \renewcommand*{\thanksmarkstyle}[1]{
224 \renewcommand{\thanksfootmark}{%
225 \thanksscript{\tmark}%
226 }
227
228 % \newlength{\thanksmarksep}

```

§ 382.6 Abstracts

```

229 \renewcommand*{\abstractcol}{
230 \renewcommand*{\abstractintoc}{
231 \renewcommand*{\abstractnum}{
232 \renewcommand*{\abstractrunin}{

```

§ 382.7 Document divisions

```

233
234 \def\@apppage{%
235   \part*{\appendixpagename}
236 }
237 \renewcommand\mempreaddappagetotochook{
238 \renewcommand\mempostaddappagetotochook{
239
240 \def\@sapppage{%
241   \part*{\appendixpagename}
242 }
243
244 \csletcs{frontmatter*}{frontmatter}
245 \csletcs{mainmatter*}{mainmatter}

```

```
246 \renewcommand*{\raggedbottomsection}{}
247 \renewcommand*{\normalbottomsection}{}
248 \renewcommand*{\bottomsectionskip}{}
249 \renewcommand*{\bottomsectionpenalty}{}
250 \csletcs{appendixpage*}{appendixpage}
251 \renewcommand*{\namedsubappendices}{}
252 \renewcommand*{\unnamedsubappendices}{}
253 \renewcommand*{\setsecnumdepth}[1]{}% todo tocvsec2
254 \renewcommand*{\maxsecnumdepth}[1]{}% todo tocvsec2
255 \renewcommand*{\beforebookskip}{}
256 \renewcommand*{\afterbookskip}{}
257 \renewcommand*{\beforepartskip}{}
258 \renewcommand*{\afterpartskip}{}
259 \renewcommand*{\midbookskip}{}
260 \renewcommand*{\midpartskip}{}
261 \renewcommand*{\printbookname}{}
262 \renewcommand*{\booknamefont}{}
263 \renewcommand*{\booknamenum}{}
264 \renewcommand*{\printbooknum}{}
265 \renewcommand*{\booknumfont}{}
266 \renewcommand*{\printpartname}{}
267 \renewcommand*{\partnamefont}{}
268 \renewcommand*{\partnamenum}{}
269 \renewcommand*{\printpartnum}{}
270 \renewcommand*{\partnumfont}{}
271 \renewcommand*{\printbooktitle}[1]{}
272 \renewcommand*{\booktitlefont}{}
273 \renewcommand*{\printparttitle}[1]{}
274 \renewcommand*{\parttitlefont}{}
275 \renewcommand*{\bookpageend}{}
276 \renewcommand*{\bookblankpage}{}
277 \renewcommand*{\nobookblankpage}{}
278 \renewcommand*{\partpageend}{}
279 \renewcommand*{\partblankpage}{}
280 \renewcommand*{\nopartblankpage}{}
281 \RenewDocumentCommand{\newleadpage}{s o m m}{}% todo
282 \RenewDocumentCommand{\renewleadpage}{s o m m}{}% todo
283 \renewcommand*{\leadpagetoclevel}{chapter}
284
285 \renewcommand*{\openright}{}
286 \renewcommand*{\openleft}{}
287 \renewcommand*{\openany}{}
288 \renewcommand*{\clearforchapter}{}
289 \renewcommand*{\memendofchapterhook}{}
290 \renewcommand*{\chapterheadstart}{}
291 % \newlength{\beforechapskip}
292 \renewcommand*{\afterchapternum}{}
293 % \newlength{\midchapskip}
294 \renewcommand*{\afterchaptertitle}{}
295 % \newlength{\afterchapskip}
```

```

296 \renewcommand*{\printchaptername}{}
297 \renewcommand*{\chapnamefont}{}
298 \renewcommand*{\chapternamenum}{}
299 \renewcommand*{\printchapternum}{}
300 \renewcommand*{\chapnumfont}{}
301 \renewcommand{\printchaptertitle}[1]{}
302 \renewcommand*{\chapttitlefont}{}
303 \renewcommand*{\printchapternonum}{}
304 \renewcommand*{\indentafterchapter}{}
305 \renewcommand*{\noindentafterchapter}{}
306 \renewcommand*{\insertchapterspace}{}
307
308 \renewcommand*{\chapterstyle}[1]{}
309 \renewcommand{\makechapterstyle}[2]{}
310 \renewcommand*{\chapindent}{}
311 \let\chapterprecis\cftchapterprecis
312 \let\chapterprecishere\cftchapterprecishere
313 \let\chapterprecistoc\cftchapterprecistoc
314 \renewcommand*{\precisfont}{}
315 \renewcommand*{\prechapterprecis}{}
316 \renewcommand*{\postchapterprecis}{}
317 \renewcommand{\precistocetext}[1]{}
318 \renewcommand*{\precistocfont}{}
319 \renewcommand*{\precistocformat}{}
320 % \newlength{\prechapterprecisshift}
321
322 \renewcommand*{\setbeforesecskip}[1]{}
323 \renewcommand*{\setaftersecskip}[1]{}
324 \renewcommand*{\setsecindent}[1]{}
325 \renewcommand*{\setsecheadstyle}[1]{}
326 \renewcommand*{\setbeforesubsecskip}[1]{}
327 \renewcommand*{\setaftersubsecskip}[1]{}
328 \renewcommand*{\setsubsecindent}[1]{}
329 \renewcommand*{\setsubsecheadstyle}[1]{}
330 \renewcommand*{\setbeforesubsubsecskip}[1]{}
331 \renewcommand*{\setaftersubsubsecskip}[1]{}
332 \renewcommand*{\setsubsubsecindent}[1]{}
333 \renewcommand*{\setsubsubsecheadstyle}[1]{}
334 \renewcommand*{\setbeforeparaskip}[1]{}
335 \renewcommand*{\setafterparaskip}[1]{}
336 \renewcommand*{\setparaindent}[1]{}
337 \renewcommand*{\setparaheadstyle}[1]{}
338 \renewcommand*{\setbeforesubparaskip}[1]{}
339 \renewcommand*{\setaftersubparaskip}[1]{}
340 \renewcommand*{\setsubparaindent}[1]{}
341 \renewcommand*{\setsubparaheadstyle}[1]{}
342 \renewcommand{\@hangfrom}[1]{#1}
343 \renewcommand{\sethangfrom}[1]{}
344 \renewcommand{\setsecnumformat}[1]{}
345

```

```

346 \renewcommand*{\hangsecnum}{ }
347 \renewcommand*{\defaultsecnum}{ }
348
349 \renewcommand*{\sechook}{ }
350 \renewcommand{\setsechook}[1]{ }
351 \renewcommand*{\subsechook}{ }
352 \renewcommand{\setsubsechook}[1]{ }
353 \renewcommand*{\subsubsechook}{ }
354 \renewcommand{\setsubsubsechook}[1]{ }
355 \renewcommand*{\parahook}{ }
356 \renewcommand{\setparahook}[1]{ }
357 \renewcommand*{\subparahook}{ }
358 \renewcommand{\setsubparahook}[1]{ }
359
360 \RenewDocumentCommand{\plainbreak}{s m}{\begin{center}~\end{center}}
361
362 \RenewDocumentCommand{\fancybreak}{s +m}{%
363 \begin{center}#2\end{center}%
364 }
365
366 \RenewDocumentCommand{\plainfancybreak}{s m m +m}{%
367 \begin{center}#4\end{center}%
368 }
369
370 \RenewDocumentCommand{\pfbreak}{s}{%
371 \begin{center}
372 \pfbreakdisplay
373 \end{center}
374 }
375
376 % \newlength{\pfbreakskip}
377 \renewcommand{\pfbreakdisplay}{*\quad*\quad*}
378
379 \renewcommand{\makeheadstyles}[2]{ }
380 \renewcommand*{\headstyles}[1]{ }

```

§ 382.8 **Pagination and headers**

```

381 \renewcommand*{\savepagenumber}{ }
382 \renewcommand*{\restorepagenumber}{ }
383 \renewcommand*{\uppercaseheads}{ }
384 \renewcommand*{\nouppercaseheads}{ }
385
386 \renewcommand*{\bookpagemark}[1]{ }
387 \renewcommand*{\partmark}[1]{ }
388 \renewcommand*{\bibmark}{ }
389 \renewcommand*{\indexmark}{ }
390 \renewcommand*{\glossarymark}{ }
391

```

```

392 \LWR@origpagestyle{empty}
393 \renewcommand*{\ps@empty}{}
394 \renewcommand*{\makepagestyle}[1]{}
395 \renewcommand*{\emptypshook}{}%
396 % \renewcommand*{\empty@oddhead}{}
397 % \renewcommand*{\empty@oddfoot}{}
398 % \renewcommand*{\empty@evenhead}{}
399 % \renewcommand*{\empty@evenfoot}{}
400 \renewcommand*{\@oddhead}{}
401 \renewcommand*{\@oddfoot}{}
402 \renewcommand*{\@evenhead}{}
403 \renewcommand*{\@evenfoot}{}
404 \renewcommand*{\aliaspagestyle}[2]{}
405 \renewcommand*{\copypagestyle}[2]{}
406
407 \renewcommand*{\makeevenhead}[4]{}
408 \renewcommand*{\makeoddhead}[4]{}
409 \renewcommand*{\makeevenfoot}[4]{}
410 \renewcommand*{\makeoddfoot}[4]{}
411 \renewcommand*{\makerunningwidth}[3]{}
412 % \newlength{\headwidth}
413 \renewcommand*{\makeheadrule}[3]{}
414 \renewcommand*{\makefootrule}[3]{}
415 \renewcommand*{\makeheadfootruleprefix}[3]{}
416 % \newlength{\normalrulethickness}
417 % \setlength{\normalrulethickness}{.4pt}
418 % \newlength{\footruleheight}
419 % \newlength{\footruleskip}
420 \renewcommand*{\makeheadposition}[5]{}
421 \renewcommand*{\makepsmarks}[2]{}
422 \renewcommand*{\makeheadfootstrut}[3]{}
423
424 \renewcommand*{\createplainmark}[3]{}
425 \renewcommand*{\memUChad}[1]{}
426 \renewcommand*{\createmark}[5]{}
427 \renewcommand*{\clearplainmark}[1]{}
428 \renewcommand*{\clearmark}[1]{}
429 \renewcommand*{\addtopsmarks}[3]{}
430 \renewcommand*{\ifonlyfloats}[2]{#2}
431 \renewcommand*{\mergepagefloatstyle}[3]{}
432
433 \renewcommand*{\framepichead}{}
434 \renewcommand*{\framepictextfoot}{}
435 \renewcommand*{\framepichook}{}
436 \renewcommand*{\showheadfootlocoff}{}
437 \renewcommand*{\showtextblocklocoff}{}

```

§ 382.9 Paragraphs and lists

```

438 \renewcommand{\hangfrom}[1]{#1}
439 \let\centerfloat\centering
440 \renewcommand*{\raggedyright}[1][]{\}
441 % \newlength{\ragrparindent}
442 \renewcommand{\sourceatright}[2][\]{\attribution{#2}}
443 \let\memorigdbs\LWR@endofline
444 \let\memorigpar\par
445 \let\atcentercr\LWR@endofline
446
447 \renewcommand*{\linenottooshort}[1][\]{\}
448 \renewcommand*{\russianpar}{\}
449 \renewcommand*{\lastlinerulefill}{\}
450 \renewcommand*{\lastlineparrule}{\}
451 \renewcommand*{\justlastaggedleft}{\}
452 \renewcommand*{\raggedrightthenleft}{\}
453 \renewcommand*{\leftcenterright}{\}
454
455 \renewcommand{\leftspringright}[4]{%
456 \begin{minipage}{#1\linewidth}\end{minipage}\quad%
457 \begin{minipage}{#2\linewidth}\begin{flushright}\end{flushright}\end{minipage}%
458 }
459
460 \renewenvironment*{blockdescription}
461 {\LWR@descriptionstart\LWR@origdescription}
462 {\enddescription}
463 \renewcommand*{\blockdescriptionlabel}[1]{\textbf{#1}}
464 \renewenvironment*{labelled}[1]{\begin{description}}{\end{description}}
465 \renewenvironment*{flexlabelled}[6]{\begin{description}}{\end{description}}
466 \renewcommand*{\tightlists}{\}
467 \renewcommand*{\defaultlists}{\}
468 \RenewDocumentCommand{\firmlists}{s}{\}
469 \renewcommand*{\firmlist}{\}
470 \renewcommand*{\tightlist}{\}
471 \renewcommand*{\zerotrivseps}{\}
472 \renewcommand*{\savetrivseps}{\}
473 \renewcommand*{\restoretrivseps}{\}

```

§ 382.10 Contents lists

```

474 \csletcs{tableofcontents*}{tableofcontents}
475 \csletcs{listoffigures*}{listoffigures}
476 \csletcs{listoftables*}{listoftables}
477 \renewenvironment{KeepFromToc}{\}{}
478 \renewcommand*{\onecoltocetc}{\}
479 \renewcommand*{\twocoltocetc}{\}
480 \renewcommand*{\ensureonecol}{\}
481 \renewcommand*{\restorefromonecol}{\}
482 \renewcommand*{\doccoltocetc}{\}
483 \renewcommand*{\maxtocdepth}[1]{\} % tocvsec2

```

```

484 \renewcommand*{\settocdepth}[1]{}% tocvsec2
485
486 \renewcommand{\toheadstart}{}
487 \renewcommand{\printtoctitle}[1]{%
488 \renewcommand{\tocmark}{}
489 \renewcommand{\aftertoctitle}{}
490 \renewcommand{\lofheadstart}{}
491 \renewcommand{\printloftitle}[1]{%
492 \renewcommand{\lofmark}{}
493 \renewcommand{\afterloftitle}{}
494 \renewcommand{\lotheadstart}{}
495 \renewcommand{\printlottitle}[1]{%
496 \renewcommand{\lotmark}{}
497 \renewcommand{\afterlottitle}{}
498
499 \renewcommand*{\setpnumwidth}[1]{%
500 \renewcommand*{\setrmarg}[1]{%
501 \renewcommand*{\cftbookbreak}{}
502 \renewcommand*{\cftpartbreak}{}
503 \renewcommand*{\cftchapterbreak}{}

504 % \newlength{\cftbeforebookskip}
505 % \newlength{\cftbookindent}
506 % \newlength{\cftbooknumwidth}
507 \renewcommand*{\cftbookfont}{}
508 \renewcommand*{\cftbookname}{}
509 \renewcommand*{\cftbookpresnum}{}
510 \renewcommand*{\cftbookaftersnum}{}
511 \renewcommand*{\cftbookaftersnumb}{}
512 \renewcommand*{\cftbookleader}{}
513 \renewcommand*{\cftbookdotsep}{1}
514 \renewcommand*{\cftbookpagefont}{}
515 \renewcommand*{\cftbookafterpnum}{}
516 \renewcommand*{\cftbookformatpnum}[1]{%
517 \renewcommand*{\cftbookformatpnumhook}[1]{%

```

Part is already defined by **tocloft**.

```

518 % \newlength{\cftbeforechapterskip}
519 % \newlength{\cftchapterindent}
520 % \newlength{\cftchapternumwidth}
521 \renewcommand*{\cftchapterfont}{}
522 \renewcommand*{\cftchaptername}{}
523 \renewcommand*{\cftchapterpresnum}{}
524 \renewcommand*{\cftchapteraftersnum}{}
525 \renewcommand*{\cftchapteraftersnumb}{}
526 \renewcommand*{\cftchapterleader}{}
527 \renewcommand*{\cftchapterdotsep}{1}
528 \renewcommand*{\cftchapterpagefont}{}
529 \renewcommand*{\cftchapterafterpnum}{}

```

```

530 \renewcommand*{\cftchapterformatpnum}[1]{ }
531 \renewcommand*{\cftchapterformatpnumhook}[1]{ }

532 % \newlength{\cftbeforesections skip}
533 % \newlength{\cftsection indent}
534 % \newlength{\cftsection numwidth}
535 \renewcommand*{\cftsection font}{ }
536 \renewcommand*{\cftsection name}{ }
537 \renewcommand*{\cftsection presnum}{ }
538 \renewcommand*{\cftsection aftersnum}{ }
539 \renewcommand*{\cftsection aftersnumb}{ }
540 \renewcommand*{\cftsection leader}{ }
541 \renewcommand*{\cftsection dotsep}{1}
542 \renewcommand*{\cftsection pagefont}{ }
543 \renewcommand*{\cftsection afterpnum}{ }
544 \renewcommand*{\cftsection formatpnum}[1]{ }
545 \renewcommand*{\cftsection formatpnumhook}[1]{ }

546 % \newlength{\cftbeforesubsections skip}
547 % \newlength{\cftsubsection indent}
548 % \newlength{\cftsubsection numwidth}
549 \renewcommand*{\cftsubsection font}{ }
550 \renewcommand*{\cftsubsection name}{ }
551 \renewcommand*{\cftsubsection presnum}{ }
552 \renewcommand*{\cftsubsection aftersnum}{ }
553 \renewcommand*{\cftsubsection aftersnumb}{ }
554 \renewcommand*{\cftsubsection leader}{ }
555 \renewcommand*{\cftsubsection dotsep}{1}
556 \renewcommand*{\cftsubsection pagefont}{ }
557 \renewcommand*{\cftsubsection afterpnum}{ }
558 \renewcommand*{\cftsubsection formatpnum}[1]{ }
559 \renewcommand*{\cftsubsection formatpnumhook}[1]{ }

560 % \newlength{\cftbeforesubsubsections skip}
561 % \newlength{\cftsubsubsection indent}
562 % \newlength{\cftsubsubsection numwidth}
563 \renewcommand*{\cftsubsubsection font}{ }
564 \renewcommand*{\cftsubsubsection name}{ }
565 \renewcommand*{\cftsubsubsection presnum}{ }
566 \renewcommand*{\cftsubsubsection aftersnum}{ }
567 \renewcommand*{\cftsubsubsection aftersnumb}{ }
568 \renewcommand*{\cftsubsubsection leader}{ }
569 \renewcommand*{\cftsubsubsection dotsep}{1}
570 \renewcommand*{\cftsubsubsection pagefont}{ }
571 \renewcommand*{\cftsubsubsection afterpnum}{ }
572 \renewcommand*{\cftsubsubsection formatpnum}[1]{ }
573 \renewcommand*{\cftsubsubsection formatpnumhook}[1]{ }

574 % \newlength{\cftbeforeparagraph skip}
575 % \newlength{\cftparagraph indent}

```

```

576 % \newlength{\cftpagraphnumwidth}
577 \renewcommand*{\cftpagraphfont}{}
578 \renewcommand*{\cftpagraphname}{}
579 \renewcommand*{\cftpagraphpresnum}{}
580 \renewcommand*{\cftpagraphaftersnum}{}
581 \renewcommand*{\cftpagraphaftersnumb}{}
582 \renewcommand*{\cftpagraphleader}{}
583 \renewcommand*{\cftpagraphdotsep}{1}
584 \renewcommand*{\cftpagraphpagefont}{}
585 \renewcommand*{\cftpagraphafterpnum}{}
586 \renewcommand*{\cftpagraphformatpnum}[1]{}
587 \renewcommand*{\cftpagraphformatpnumhook}[1]{}

588 % \newlength{\cftbeforesubparagraphskip}
589 % \newlength{\cftsubparagraphindent}
590 % \newlength{\cftsubparagraphnumwidth}
591 \renewcommand*{\cftsubparagraphfont}{}
592 \renewcommand*{\cftsubparagraphname}{}
593 \renewcommand*{\cftsubparagraphpresnum}{}
594 \renewcommand*{\cftsubparagraphaftersnum}{}
595 \renewcommand*{\cftsubparagraphaftersnumb}{}
596 \renewcommand*{\cftsubparagraphleader}{}
597 \renewcommand*{\cftsubparagraphdotsep}{1}
598 \renewcommand*{\cftsubparagraphpagefont}{}
599 \renewcommand*{\cftsubparagraphafterpnum}{}
600 \renewcommand*{\cftsubparagraphformatpnum}[1]{}
601 \renewcommand*{\cftsubparagraphformatpnumhook}[1]{}

602 % \newlength{\cftbeforefigureskip}
603 % \newlength{\cftfigureindent}
604 % \newlength{\cftfigurenumwidth}
605 \renewcommand*{\cftfigurefont}{}
606 \renewcommand*{\cftfigurename}{}
607 \renewcommand*{\cftfigurepresnum}{}
608 \renewcommand*{\cftfigureaftersnum}{}
609 \renewcommand*{\cftfigureaftersnumb}{}
610 \renewcommand*{\cftfigureleader}{}
611 \renewcommand*{\cftfiguredotsep}{1}
612 \renewcommand*{\cftfigurepagefont}{}
613 \renewcommand*{\cftfigureafterpnum}{}
614 \renewcommand*{\cftfigureformatpnum}[1]{}
615 \renewcommand*{\cftfigureformatpnumhook}[1]{}

616 % \newlength{\cftbeforesubfigureskip}
617 % \newlength{\cftsubfigureindent}
618 % \newlength{\cftsubfigurenumwidth}
619 \newcommand*{\cftsubfigurefont}{}
620 \newcommand*{\cftsubfigurename}{}
621 \newcommand*{\cftsubfigurepresnum}{}
622 \newcommand*{\cftsubfigureaftersnum}{}

```

```

623 \newcommand*{\cftsubfigureaftersnumb}{}
624 \newcommand*{\cftsubfigurereleader}{}
625 \newcommand*{\cftsubfiguredotsep}{1}
626 \newcommand*{\cftsubfigurepagefont}{}
627 \newcommand*{\cftsubfigureafterpnum}{}
628 \newcommand*{\cftsubfigureformatpnum}[1]{}
629 \newcommand*{\cftsubfigureformatpnumhook}[1]{}

630 % \newlength{\cftbeforetableskip}
631 % \newlength{\cfttableindent}
632 % \newlength{\cfttablenumwidth}
633 \renewcommand*{\cfttablefont}{}
634 \renewcommand*{\cfttablename}{}
635 \renewcommand*{\cfttablepresnum}{}
636 \renewcommand*{\cfttableaftersnum}{}
637 \renewcommand*{\cfttableaftersnumb}{}
638 \renewcommand*{\cfttableleleader}{}
639 \renewcommand*{\cfttabledotsep}{1}
640 \renewcommand*{\cfttablepagefont}{}
641 \renewcommand*{\cfttableafterpnum}{}
642 \renewcommand*{\cfttableformatpnum}[1]{}
643 \renewcommand*{\cfttableformatpnumhook}[1]{}

644 % \newlength{\cftbeforesubtableskip}
645 % \newlength{\cftsubtableindent}
646 % \newlength{\cftsubtablenumwidth}
647 \newcommand*{\cftsubtablefont}{}
648 \newcommand*{\cftsubtablename}{}
649 \newcommand*{\cftsubtablepresnum}{}
650 \newcommand*{\cftsubtableaftersnum}{}
651 \newcommand*{\cftsubtableaftersnumb}{}
652 \newcommand*{\cftsubtableleleader}{}
653 \newcommand*{\cftsubtabledotsep}{1}
654 \newcommand*{\cftsubtablepagefont}{}
655 \newcommand*{\cftsubtableafterpnum}{}
656 \newcommand*{\cftsubtableformatpnum}[1]{}
657 \newcommand*{\cftsubtableformatpnumhook}[1]{}

658 \renewcommand*{\booknumberline}[1]{}
659 \renewcommand*{\partnumberline}[1]{}
660 \renewcommand*{\chapternumberline}[1]{}
661 \renewcommand*{\numberlinehook}[1]{}
662 % \renewcommand*{\cftwhatismyname}{}%
663 \renewcommand*{\booknumberlinehook}[1]{}
664 \renewcommand*{\partnumberlinehook}[1]{}
665 \renewcommand*{\chapternumberlinehook}[1]{}
666 \renewcommand{\numberlinebox}[2]{}
667 \renewcommand{\booknumberlinebox}[2]{}
668 \renewcommand{\partnumberlinebox}[2]{}
669 \renewcommand{\chapternumberlinebox}[2]{}

```

```

670 %
671 % \newlength{\cftparfillskip}
672 \renewcommand*{\cftpagenumbersoff}[1]{%
673 \renewcommand*{\cftpagenumberon}[1]{%
674 \renewcommand*{\cftlocalchange}[3]{%
675 \renewcommand*{\cftaddtitleline}[4]{%
676 \renewcommand*{\cftaddnumtitleline}[4]{%
677 \renewcommand{\cftinsertcode}[2]{%
678 \renewcommand{\cftinserthook}[2]{%
679 \renewcommand{\settocpreprocessor}[2]{%
680 \DeclareRobustCommand{\cftpagenumbersoff}[1]{%
681 \DeclareRobustCommand{\cftpagenumberon}[1]{%

```

§ 382.11 Floats and captions

`\newfloat` [*<1: within>*] [*<2: type>*] [*<3: ext>*] [*<4: capname>*]

```

682 \RenewDocumentCommand{\newfloat}{o m m m}{%
683 \IfValueTF{#1}%
684 {\DeclareFloatingEnvironment[fileext=#3,within=#1,name={#4}]{#2}}%
685 {\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.

```

686 \cslet{listof#2s}\relax%
687 \cslet{listof#2es}\relax%
688 }

```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]

Emulated through the `\newfloat` mechanism. Note that **memoir** uses a different syntax than **tocloft** for the name.

```

689 \RenewDocumentCommand{\newlistof}{o m m m}{
690 {%
691 \IfValueTF{#1}
692 {\newlistentry[#1]{#2}{#3}{0}}
693 {\newlistentry{#2}{#3}{0}}
694 \@namedef{ext@#2}{#3}%
695 \@ifundefined{c@#3depth}{\newcounter{#3depth}}{}%
696 \setcounter{#3depth}{1}%
697 \@namedef{#3mark}{}%
698 \@namedef{#2}{\listof{#2}{#4}}
699 \@namedef{@cftmake#3title}{}
700 \@ifundefined{cftbefore#3titleskip}{
701 \expandafter\newlength\csname cftbefore#3titleskip\endcsname
702 \expandafter\newlength\csname cftafter#3titleskip\endcsname
703 }{}
704 \@namedef{cft#3titlefont}{}
705 \@namedef{cftafter#3title}{}

```

```

706 \@namedef{cft#3prehook}{}
707 \@namedef{cft#3posthook}{}
708 }

```

```

709 \renewcommand{\setfloatadjustment}[2]{}

```

Borrowed from the **lwarp** version of **keyfloat**:

```

710 \NewDocumentEnvironment{KFLTmemoir@marginfloat}[0]{-1.2ex}{m}
711 {% start
712 \LWR@BlockClassWP{float:right; width:2in; margin:10pt}{}{marginblock}%
713 \captionsetup{type=#2}%
714 }
715 {%
716 \endLWR@BlockClassWP%
717 }
718
719 \DeclareDocumentEnvironment{marginfigure}[o]
720 {\begin{KFLTmemoir@marginfloat}{figure}}
721 {\end{KFLTmemoir@marginfloat}}
722
723 \DeclareDocumentEnvironment{margintable}[o]
724 {\begin{KFLTmemoir@marginfloat}{table}}
725 {\end{KFLTmemoir@marginfloat}}

726 \renewcommand{\setmarginfloatcaptionadjustment}[2]{}
727 \renewcommand{\setmpjustification}[2]{}
728 \renewcommand*{\mpjustification}{}
729 \renewcommand*{\setfloatlocations}[2]{}
730 \DeclareDocumentCommand{\suppressfloats}[o]{}
731 \renewcommand*{\FloatBlock}{}
732 \renewcommand*{\FloatBlockAllowAbove}{}
733 \renewcommand*{\FloatBlockAllowBelow}{}
734 \renewcommand*{\setFloatBlockFor}{}
735
736 \renewcommand{\captiontitlefinal}[1]{}
737
738 \renewcommand{\flegtable}{\tablename}
739 \renewcommand{\flegfigure}{\figurename}
740 \renewcommand{\flegtocfigure}{\tableofcontents}
741 \renewcommand{\flegtocfigure}{}
742
743
744 \renewcommand{\subcaption}[2][{}]{%
745 \ifblank{#1}{\subfloat[#2]{}{}{\subfloat[#1][#2]{}{}}}%
746 }
747
748 \renewcommand{\contsubcaption}{\ContinuedFloat\subcaption}
749

```

```

750 \LetLtxMacro\subcaptionref\subref
751
752 \renewcommand*{\tightsubcaptions}{}
753 \renewcommand*{\loosesubcaptions}{}
754
755 \renewcommand*{\subcaptionsize}[1]{}
756 \renewcommand*{\subcaptionlabelfont}[1]{}
757 \renewcommand*{\subcaptionfont}[1]{}
758 \renewcommand*{\subcaptionstyle}[1]{}
759
760 \renewcommand*{\hangsubcaption}{}
761 \renewcommand*{\shortsubcaption}{}
762 \renewcommand*{\normalsubcaption}{}
763
764 \RenewDocumentEnvironment{sidecaption}{o m o}
765 {}
766 {
767 \IfValueTF{#1}{\caption[#1]{#2}}{\caption{#2}}%
768 \IfValueT{#3}{\label{#3}}%
769 }
770
771 % \newlength{\sidecapwidth}
772 % \newlength{\sidecapsep}
773 \renewcommand*{\setsidecaps}[2]{}
774 \renewcommand*{\sidecapmargin}[1]{}
775 % \newif\ifscapmargleft
776 \scapmargleftfalse
777 \renewcommand*{\setsidecappos}[1]{}
778
779 \RenewDocumentEnvironment{sidecontcaption}{m o}
780 {}
781 {%
782 \ContinuedFloat%
783 \caption{#1}%

```

Without \@captype, the section is referred to instead.

```

784 \IfValueT{#2}{\label[\@captype]{#2}}%
785 }

```

\sidenamedlegend does not appear to use the TOC argument.

```

786 \renewenvironment{sidenamedlegend}[2][]{
787 \begin{center}
788 \csuse{\@captype name}\CaptionSeparator#2
789 \end{center}
790 }
791 {}
792

```

```

793 \renewenvironment{sidelegend}[1]
794 {\begin{center}
795 #1
796
797 }
798 {\end{center}}
799
800 \renewcommand*{\sidecapstyle}{}
801 \renewcommand*{\overridescapmargin}[1]{}
802 % \newlength{\sidecapraise}
803 \renewcommand*{\sidecapfloatwidth}{\linewidth}
804
805 \LetLtxMacro\ctabular\tabular
806 \LetLtxMacro\endctabular\endtabular
807
808 \renewcommand{\autorows}[5][\%
809 #5
810 }
811
812 \renewcommand{\autocol}[5][\%
813 #5
814 }

```

§ 382.12 Page notes

```

815 \renewcommand*{\feetabovelfloat}{}
816 \renewcommand*{\feetbelowfloat}{}
817 \renewcommand*{\feetatbottom}{}
818
819 \renewcommand*{\verbfootnote}[2][{}
820 \PackageError{lwarp, memoir}
821 {Verbatim footnotes are not yet supported by lwarp.}
822 {This may be improved some day.}
823 }
824
825 \renewcommand*{\plainfootnotes}{}
826 \renewcommand*{\twocolumnfootnotes}{}
827 \renewcommand*{\threecolumnfootnotes}{}
828 \renewcommand*{\paragraphfootnotes}{}
829 \renewcommand*{\footfudgefiddle}{}
830
831 \renewcommand*{\newfootnoteseries}[1]{
832 \PackageError{lwarp, memoir}
833 {Memoir footnote series are not yet supported by lwarp.}
834 {This may be improved some day.}
835 }
836
837 \renewcommand*{\plainfootstyle}[1]{}
838 \renewcommand*{\twocolumnfootstyle}[1]{}

```

```
839 \renewcommand*{\threecolumnfootstyle}[1]{}
840 \renewcommand*{\paragraphfootstyle}[1]{}
841
842 \renewcommand*{\footfootmark}{}
843 \renewcommand*{\footmarkstyle}[1]{}
844
845 % \newlength{\footmarkwidth}
846 % \newlength{\footmarksep}
847 % \newlength{\footparindent}
848
849 \renewcommand*{\foottextfont}{}
850
851 \renewcommand*{\marginparmargin}[1]{}
852 \renewcommand*{\sideparmargin}[1]{}
853
854 \LetLtxMacro\sidepar\marginpar
855 \renewcommand*{\sideparfont}{}
856 \renewcommand*{\sideparform}{}
857 \LWR@providelength{\sideparvshift}
858
859 \renewcommand*{\parnopar}{}
860
861 \renewcommand{\sidebar}[1]{\begin{quote}#1\end{quote}}
862 \renewcommand*{\sidebarmargin}[1]{}
863 \renewcommand*{\sidebarfont}{}
864 \renewcommand*{\sidebarform}{}
865 % \newlength{\sidebarhsep}
866 % \newlength{\sidebarvsep}
867 % \newlength{\sidebarwidth}
868 % \newlength{\sidebartopsep}
869 \renewcommand{\setsidebarheight}[1]{}
870 \renewcommand*{\setsidebars}[6]{}
871 \renewcommand*{\footnotesatfoot}{}
872 \renewcommand*{\footnotesinmargin}{}
873
874 \LetLtxMacro\sidefootnote\footnote
875 \LetLtxMacro\sidefootnotemark\footnotemark
876 \LetLtxMacro\sidefootnotetext\footnotetext
877
878 \renewcommand*{\sidefootmargin}[1]{}
879 % \newlength{\sidefoothsep}
880 % \newlength{\sidefootvsep}
881 % \newlength{\sidefootwidth}
882 % \newlength{\sidefootadjust}
883 % \newlength{\sidefootheight}
884 \renewcommand*{\setsidefootheight}[1]{}
885 % \renewcommand*{\sidefootfont}{}% in docs but not in the package
886 \renewcommand*{\setsidefeet}[6]{}
887 \renewcommand*{\sidefootmarkstyle}[1]{}
888 \renewcommand*{\sidefoottextfont}{}

```

```

889 \renewcommand*{\sidefootform}{}
890
891 \renewcommand*{\continuousnotenums}{\pncontopttrue}% from pagenote
892 \renewcommand*{\notepageref}{}
893 \renewcommand*{\prenotetext}{}
894 \renewcommand*{\postnotetext}{}
895 \renewcommand*{\idtextinnotes}[1]{}
896 \renewcommand*{\printpageinnotes}[1]{}
897 \renewcommand*{\printpageinnoteshyperref}[1]{}
898 \renewcommand*{\foottopagenote}{}
899 \renewcommand*{\pagetofootnote}{}

```

§ 382.13 Decorative text

```

900 \renewcommand*{\epigraphposition}[1]{}
901 \renewcommand*{\epigraphtextposition}[1]{}
902 \renewcommand*{\epigraphsourceposition}[1]{}
903 \renewcommand*{\epigraphfontsize}[1]{}
904 \renewcommand*{\epigraphforheader}[2] [] {}
905 \renewcommand*{\epigraphpicture}{}

```

§ 382.14 Poetry

```

906 \renewcommand*{\vinphantom}{}
907 \renewcommand*{\vleftoffline}[1]{#1}
908 % \let\linenumberfrequency\poemlines
909 % \renewcommand*{\linenumberfont}[1]{}
910
911 \DeclareDocumentCommand{\PoemTitle}{s o m}{%
912 \IfValueTF{#2}%
913 {\poemtitle{#2}{#4}}%
914 {\poemtitle{#4}}%
915 }
916
917 \renewcommand*{\NumberPoemTitle}{}
918 \renewcommand*{\PlainPoemTitle}{}
919 \renewcommand*{\poemtitlepstyle}{}
920 \renewcommand*{\poemtitlestarmark}[1]{}
921 \renewcommand*{\poemtitlestarpstyle}{}
922 \renewcommand*{\PoemTitleheadstart}{}
923 \renewcommand*{\printPoemTitlenonum}{}
924 \renewcommand*{\printPoemTitlenum}{}
925 \renewcommand*{\afterPoemTitlenum}{}
926 \renewcommand*{\printPoemTitledtitle}[1]{}
927 \renewcommand*{\afterPoemTitle}{}
928 \newlength{\midpoemtitleskip}
929 \renewcommand*{\PoemTitlenumfont}{}
930 \renewcommand*{\PoemTitlefont}{}

```

§ 382.15 Boxes, verbatims and files

```
931 \renewenvironment{qframe}{\framed}{\endframed}
932 \renewenvironment{qshade}{\shaded}{\endshaded}
```

Use the **comment** package:

```
933 \renewcommand*{\commentsoff}[1]{\includecomment{#1}}
934 \renewcommand*{\commentson}[1]{\excludecomment{#1}}
935 \LetLtxMacro\renewcomment\commentson
936
937 \renewcommand*{\setverbatimfont}[1]{\fontfamily{#1}}
938 \renewcommand*{\tabson}[1]{\fontfamily{#1}}
939 \renewcommand*{\tabsoff}{}
940 \renewcommand*{\wrappingon}{}
941 \renewcommand*{\wrappingoff}{}
942 \renewcommand*{\verbatimindent}{}
943 \renewcommand*{\verbatimbreakchar}[1]{\fontfamily{#1}}
944 \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.

```
945 \renewcommand*{\bvbox}{}
946 \renewcommand*{\bvttopandtail}{}
947 \renewcommand*{\bvsides}{}
948 \renewcommand*{\nobvbox}{}
949 % \newlength\bvboxsep
950 \renewcommand*{\bvtoprulehook}{}
951 \renewcommand*{\bvtopmidhook}{}
952 \renewcommand*{\bvendrulehook}{}
953 \renewcommand*{\bvleftsidehook}{}
954 \renewcommand*{\bvrightsidehook}{}
955 \renewcommand*{\bvperpagetrue}{}
956 \renewcommand*{\bvperpagefalse}{}
957 \renewcommand*{\bvtopofpage}[1]{\fontfamily{#1}}
958 \renewcommand*{\bvendofpage}[1]{\fontfamily{#1}}
959 \renewcommand*{\linenumberfrequency}[1]{\fontfamily{#1}}
960 \renewcommand*{\resetbvlinenumber}{}
961 \renewcommand*{\setbvlinenums}[2]{\fontfamily{#1}}
962 \renewcommand*{\linenumberfont}[1]{\fontfamily{#1}}
963 \renewcommand*{\bvnumbersinside}{}
964 \renewcommand*{\bvnumbersoutside}{}

```

§ 382.16 Cross referencing

```
965 \renewcommand*{\fref}[1]{\cref{#1}}
966 \renewcommand*{\tref}[1]{\cref{#1}}
```

```

967 \renewcommand*{\pref}[1]{\cpageref{#1}}
968 \renewcommand*{\Aref}[1]{\cref{#1}}
969 \renewcommand*{\Bref}[1]{\cref{#1}}
970 \renewcommand*{\Pref}[1]{\cref{#1}}
971 \renewcommand*{\Sref}[1]{\cref{#1}}
972 \renewcommand*{\figurerefname}{Figure}
973 \renewcommand*{\tablerefname}{Table}
974 \renewcommand*{\pagerefname}{page}
975 \renewcommand*{\bookrefname}{Book~}
976 \renewcommand*{\partrefname}{Part~}
977 \renewcommand*{\chapterrefname}{Chapter~}
978 \renewcommand*{\sectionrefname}{S}
979 \renewcommand*{\appendixrefname}{Appendix~}
980 \LetLtxMacro\titleref\nameref
981 \renewcommand*{\headnameref}{}
982 \renewcommand*{\tocnameref}{}
983
984 \providecounter{LWR@currenttitle}
985
986 \renewcommand*{\currenttitle}{%
987   \addtocounter{LWR@currenttitle}{1}%
988   \label{currenttitle\arabic{LWR@currenttitle}}%
989   \nameref{currenttitle\arabic{LWR@currenttitle}}%
990 }
991
992 \renewcommand*{\theTitleReference}[2]{}
993 \renewcommand*{\namerefon}{}
994 \renewcommand*{\namerefoff}{}

```

§ 382.17 Back matter

```

995 \DeclareDocumentCommand{\newblock}{}{}
996 %
997 \renewcommand*{\showindexmarks}{}
998 \renewcommand*{\hideindexmarks}{}
999
1000 \renewcommand*{\xindyindex}{}

```

§ 382.18 Miscellaneous

```

1001 \renewcommand*{\changemarks}{}
1002 \renewcommand*{\nochangemarks}{}
1003 \renewcommand*{\added}[1]{}
1004 \renewcommand*{\deleted}[1]{}
1005 \renewcommand*{\changed}[1]{}
1006
1007 \renewcommand*{\showtrimsoff}{}
1008 \renewcommand*{\showtrimson}{}
1009 \renewcommand*{\trimXmarks}{}

```

```

1010 \renewcommand*{\trimLmarks}{}
1011 \renewcommand*{\trimFrame}{}
1012 \renewcommand*{\trimNone}{}
1013 \renewcommand*{\trimmarkscolor}{}
1014 \renewcommand*{\trimmarks}{}
1015 \renewcommand*{\tmarktl}{}
1016 \renewcommand*{\tmarktr}{}
1017 \renewcommand*{\tmarkbr}{}
1018 \renewcommand*{\tmarkbl}{}
1019 \renewcommand*{\tmarktm}{}
1020 \renewcommand*{\tmarkmr}{}
1021 \renewcommand*{\tmarkbm}{}
1022 \renewcommand*{\tmarkml}{}
1023 \renewcommand*{\trimmark}{}
1024 \renewcommand*{\quarkmarks}{}
1025 \renewcommand*{\registrationColour}[1]{}
1026
1027 \renewcommand*{\leavespergathering}[1]{}
1028
1029 \renewcommand*{\noprelistbreak}{}
1030
1031 \renewcommand*{\cleartorecto}{}
1032 \renewcommand*{\cleartoverso}{}
1033
1034 \renewenvironment{vplace}[1][ ]{}{}

```

§ 382.19 Ccaption emulation

```

1035 \renewcommand*{\captiondelim}[1]{\renewcommand*{\CaptionSeparator}{#1}}
1036 \renewcommand*{\captionnamefont}[1]{}
1037 \renewcommand*{\captiontitlefont}[1]{}
1038 \renewcommand*{\flushleft}{}
1039 \renewcommand*{\centerlastline}{}
1040 \renewcommand*{\captionstyle}[2][ ]{}
1041 \DeclareDocumentCommand{\captionwidth}{m}{}
1042 \renewcommand*{\changecaptionwidth}{}
1043 \renewcommand*{\normalcaptionwidth}{}
1044 \renewcommand*{\hangcaption}{}
1045 \renewcommand*{\indentcaption}[1]{}
1046 \renewcommand*{\normalcaption}{}
1047 \renewcommand{\precaption}[1]{}
1048 \renewcommand{\postcaption}[1]{}
1049 \renewcommand{\midbicaption}[1]{}
1050 \renewcommand{\contcaption}[1]{%
1051 % \ContinuedFloat%
1052 % \caption{#1}%
1053 \begin{LWR@figcaption}% later becomes \caption*
1054 \csuse{\@capttype name}{\thechapter.\the\value{\@capttype}\CaptionSeparator #1
1055 \end{LWR@figcaption}

```

```

1056 }

1057 \newlength{\abovelegendskip}
1058 \setlength{\abovelegendskip}{0.5\baselineskip}
1059 \newlength{\belowlegendskip}
1060 \setlength{\belowlegendskip}{\abovelegendskip}

```

The extra `\\` here forces a `
` in HTML when `\legend` is used in a `\marginpar`.

```

1061 \renewcommand{\legend}[1]{\begin{center}#1\\end{center}}
1062
1063 \renewcommand{\namedlegend}[2][ ]{
1064 \begin{center}
1065 \csuse{fleg@captive}\CaptionSeparator#2\\
1066 \end{center}
1067 \csuse{flegtoc@captive}{#1}
1068 }
1069
1070 \renewcommand{\newfixedcaption}[3][\caption]{%
1071 \renewcommand{#2}{\def@captive{#3}#1}}
1072 \renewcommand{\renewfixedcaption}[3][\caption]{%
1073 \renewcommand{#2}{\def@captive{#3}#1}}
1074 \renewcommand{\providefixedcaption}[3][\caption]{%
1075 \providecommand{#2}{\def@captive{#3}#1}}
1076
1077 \renewcommand{\bitwonumcaption}[6][ ]{%
1078 \ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
1079 \addtocounter{@captive}{-1}%
1080 \begingroup%
1081 \csdef{@captive name}{#4}%
1082 \ifblank{#5}{\caption{#6}}{\caption[#5]{#6}}%
1083 \endgroup%
1084 \ifblank{#1}{\label{#1}}%
1085 }
1086
1087 \LetLtxMacro\bionenumcaption\bitwonumcaption% todo
1088
1089 \renewcommand{\bicaption}[5][ ]{%
1090 \ifblank{#2}{\caption{#3}}{\caption[#2]{#3}}%
1091 \begin{LWR@figcaption}% later becomes \caption*
1092 #4 \thechapter.\the\value{@captive}\CaptionSeparator #5
1093 \end{LWR@figcaption}
1094 \ifblank{#1}{\label{#1}}%
1095 }
1096
1097 \renewcommand{\bicontcaption}[3]{%
1098 \contcaption{#1}%
1099 \begingroup%
1100 \csdef{@captive name}{#2}%
1101 \contcaption{#3}%

```

```

1102 \endgroup%
1103 }

1104 % only in ccaption, not in memoir:
1105 % \LetLtxMacro\longbitwounumcaption\bitwounumcaption%
1106 % \LetLtxMacro\longbionenumcaption\bitwounumcaption%
1107 % \LetLtxMacro\longbicaption\bicaption%
1108
1109 \RenewDocumentCommand{\subtop}{0{} 0{} m}{%
1110 \subfloat[#1][#2]{#3}%
1111 }
1112
1113 \RenewDocumentCommand{\subbottom}{0{} 0{} m}{%
1114 \subfloat[#1][#2]{#3}%
1115 }
1116
1117 \renewcommand{\contsubtop}{%
1118 \ContinuedFloat\addtocounter{@capttype}{1}%
1119 \subtop}
1120
1121 \renewcommand{\contsubbottom}{%
1122 \ContinuedFloat\addtocounter{@capttype}{1}%
1123 \subbottom}
1124
1125 \renewcommand{\subconcluded}{}
1126
1127 \let\subfigure\subbottom
1128 \let\subtable\subtop
1129
1130 \let\contsubtable\contsubtop
1131 \let\contsubfigure\contsubbottom

1132 \newcommand{\newfloatentry}[4][\@empty]{TODO: newfloatentry}
1133 \newcommand{\newfloatlist}[5][\@empty]{TODO: newfloatlist}
1134 \newcommand{\newfloatenv}[4][\@empty]{TODO: newfloatenv}
1135 \DeclareRobustCommand{\newfloatpagesoff}[1]{}
1136 \DeclareRobustCommand{\newfloatpageson}[1]{}
1137 \newcommand{\setnewfloatindents}[3]{}

```

§ 382.20 Final patchwork

```

1138 \newlistof{tableofcontents}{toc}{\contentsname}
1139 \newlistof{listoffigures}{lof}{\listfigurename}
1140 \newlistof{listoftables}{lot}{\listtablename}

```

Change History and Index

For the most recent changes and the start of the Index, see page 929.

§ 382 Change History

v0.10	General: 2016/03/08 Initial version . . . 1	Docs: Table: Cross-referencing data structures. 424
v0.11	General: 2016/03/11 1	Docs: Table: Float data structures. 435
	Added section: Operating-System portability. 165	Docs: Trademarks section. 546
	Added section: Selecting the operating system. 105	Docs: Troubleshooting cross-references. 154
	Test Suite: MS-WINDOWS in README.txt 1	Test Suite: Assigned cleveref name for Test Float. 1
	Test Suite: limages and index in README.txt 1	Test Suite: Floatrow 1
v0.12	\LWR@newhtmlfile: Bugfix: TOC with numbered files. 312	v0.15
	General: 2016/03/14 1	General: 2016/04/06 1
	Global: Uses \p@(type) in float captions. 1	Added. 658
	Test Suite: Sub-figures 1	Ampersand (&): Fixed handling when passed as an argument. . . 367
v0.13	\CaptionSeparator: Fix for newer babel package. 439	Docs: Added warning icons for items needing special attention. 161
	\LWR@LwarpStart: \up and \fup . . 329	Docs: Clarify print/HTML output. 105
	General: 2016/03/24 1	Docs: Moved the supported features table to the introduction. 63
	Fix dollar-redefined bug for newer package. 816	Files: lwarp_formal.css added. . . . 1
	Removed package: subfig 1	Fix: steps counter 657
	Test Suite: Ordinals, Subcaption . . 1	Fixed & handling. 656
v0.14	\LWR@htmlsectionfilename: Fix: Links to home page. 275	Test Suite: test_suite_formal.css file added. 1
	General: 2016/03/31 1	v0.16
	floatrow : Added. 653	General: 2016/04/11 1
	Docs: Commands for a successful HTML conversion. 109	\titlingpage: Improved print-output spacing. 338
	Docs: Commands into a warpprint environment. 107	xfrac : Adjusted for the use of any font: 871
	Docs: Newclude limitations. . . . 139	Added XeLaTeX, LuaLaTeX support. 162
		Docs: Font and UTF-8 support. . . 94
		Docs: Moved location of \usepackage{lwarp}. 96
		Docs: Text not converting. 154
		Fix: amsmath options clash 182
		Fix: newtxmath compatibility. . . 182

Lwarp no longer selects fonts.	94, 178	Reorganize \HomeHTMLFilename	
Removed package: suffix	1	logic.	497
Test Suite: Improved titlingpage.	338	Surpress extra space.	497
Test Suite: Lwarp no longer selects		verse: Supports verse, memoir	
fonts.	1	packages.	850
Test Suite: Supports XeLaTeX,		minipage: Fix: \linewidth,	
LuaLaTeX.	1	\textwidth, \textheight	
v0.17		inside a minipage.	517
\LWR@htmlsectionfilename: Fix:		v0.19	
Links when entire doc is one		\HTMLFilename: Docs: Escape	
HTML page.	275	filename underscores.	274
General: 2016/04/14	1	\HomeHTMLFilename: Docs: Escape	
mdframed : Added.	717	filename underscores.	275
Test Suite: Fix: Print-version		\LWR@LwarpStart: Enabled \\ equal	
front-matter page numbers.	1	to \newline.	328
Test Suite: Mdfamed	1	\LWR@doequation: MATHJAX	
v0.18		support.	476
\LWR@includegraphicsb: Add: svgz		\LWR@doubledollar: MATHJAX	
file extension.	676	support.	470
em, ex, %, px dimensions		\LWR@filestart: lwarp_mathjax.txt	
preserved.	676	loaded.	326
Fix: \linewidth, \textwidth,		\LWR@minipagestartpars:	
\textheight inside a minipage.	676	Surpresses paragraph tags	
Improved HTML output linebreaks.	676	between minipages.	535
\LWR@myshorttoc: Reorganize		\LWR@subsingledollar: MATHJAX	
\HomeHTMLFilename logic.	444	support.	464
\LWR@newhtmlfile: sidetoc after		\LateximageFontSizeName: Add:	
title, improving responsive		User-adjustable math/lateximage	
design.	311	font size.	492
\LWR@requesttoc: Reorganize		\hspace: Fix: \hspace length	
\HomeHTMLFilename logic.	331	computations.	536
\LWR@subhyperref: Improved HTML		\minipagefullwidth: Added: No	
output linebreaks.	432	width tag for the next minipage	
\LWR@subhyperrefclass: Improved		in HTML.	517
HTML output linebreaks.	432	\warpHTMLonly: Added.	169
\LWR@subinlineimage: Surpress		\warpprintonly: Replaces	
extra space.	434	\rowprintedonly.	169
\hspace: \hspace supported.	536	\xfracHTMLfontsize: Added.	870
General: 2016/05/19	1	General: 2016/06/08	1
File: lwarp.css: Improved TOC		MATHJAX support added.	474, 480, 481
outline display.	1	multirow : Added optional args.	736
Files: lwarp.css and		Adapts to tikz version.	816
lwarp_formal.css: Improved		Avoids MATHJAX.	463
responsive design.	1	cleveref: Loaded	
Microtype disabled during HTML		\AtEndPreamble.	512
generation	179	CSS for table note item.	815
PDF Unicode input characters.	163	Docs: Math options.	96
Test Suite: Verse package	1	Docs: Table: Cross-referencing	
lateximage: pdfcrop: --hires added.	497	data structures, updated.	424

File: lwarp.css:	
tnoteitemheader added.	1
File: lwarp_mathjax.txt added. .	1
Introduction: MATHJAX support	
mentioned.	61
Options: mathsvg and mathjax .	167
Supports colored \rule.	862
titles: null \pagestyle and	
\thispagestyle for HTML. . .	817
v0.20	
\BlockClassSingle: Renamed from	
"LWR@htmldivclassline".	288
\HTMLDescription: Added	
\NewHTMLdescription.	
(Renamed in v0.30.)	299
\HTMLFilename: No longer escape	
underscores.	274
\HomeHTMLFilename: No longer	
escape underscores.	275
\InlineClass: Renamed from	
"inlineclass".	288
\LWR@LwarpStart: Fix: math cross	
references.	330
\LWR@closeparagraph: \unskip	
extra spaces.	292
No break tags in the start/end of a	
tabular.	292
\LWR@endoffline: Fix: \\	535
\LWR@filestart: Adds meta	
description.	326
\LWR@htmldivclass: Added	
optional style.	286
\LWR@htmlclass: Added	
optional style.	286
\LWR@htmlsectionfilename:	
HTMLFilename: removed	
additional trailing '-', and may be	
empty.	275
Sections called "Index" or "index"	
have an underscore prepended to	
their filenames if no prefix. . .	275
\LWR@includegraphicsb: Fix:	
\linewidth in a floatrow.	676
Fix: Expands filename.	676
\LWR@longtabledatacaptiontag:	
Fix: Pars in captions.	407
\LWR@section: Combined	
higher-level sections together	
into files.	318
\LWR@setOSWindows: Auto-detects	
operating system.	165
\LWR@subhtmlclass:	
Factored code.	285
\SetHTMLFileNumber: Add: Control	
file numbers.	275
\cpagerefFor: User-redefinable	
word for page references.	514
\dotfill: Inserts an ellipsis.	534
\hfill: Inserts a \qqquad.	534
\hrulefill: Inserts a short rule. . .	534
\hspace: Add: Supports HTML thin	
breakable space.	536
\hyperindexref: Print mode	
provided in case hyperref not	
used.	454
\pageref: Added.	431
\tracinglwarp: Added.	189
General: 2017/02/09	1
afterpage : Added.	555
alltt : Added.	560
bookmark : Added.	574
caption and subcaption	
supported.	1
cleveref and referencing patches:	
Applied \AfterEndPreamble. .	512
draftwatermark : Added.	620
eso-pic : Added.	630
everypage : Added.	631
extramarks : Added.	631
fancyhdr : Added.	638
float : Improved float caption type	
handling.	650
hyperref : Additional user macros. 688	
keyfloat : Added.	697
letterspace : User-interface	
emulated.	700
listings : Added.	704
ltxcaption : Added.	711
lwarp-newproject : Added.	202
microtype : User-interface	
emulated.	731
needspace : Added.	740
nowidow : Added.	743
placeins : Added.	763
ragged2e : Added.	766
setspace : Improved support. . .	782
textpos : Added.	810
titles : Added.	817
titlesec : Added.	821

titletoc: Added.	823	newclude: Added.	741
titling: Improved compatibility. . .	825	newunicodechar: Added.	741
tocloft: Added.	833	lwarpmk: Fix: <i>lwarpmk again</i> for	
wallpaper: Added.	855	WINDOWS.	248
wrapfig: Added.	856	lwarpmk: Fix: <i>lwarpmk limages</i>	
Added @, <, > columns.	361	for WINDOWS.	248
Added single-expansion data		lwarpmk: Fix: <i>lwarpmk</i> uses	
arrays.	270	<i>lateximages</i> text file instead of	
Code factored into independent		shell script.	248
<i>lwarp_html</i> files.	546	Add: Errors for misplaced	
Docs: Examples for generating		packages.	170
HTML file names.	103	Docs: Added internet class.	68
Docs: Improved index.	1	Docs: Added TeX2page, GladTeX.	68
Enhanced titling support.	336	Docs: Installing on WINDOWS.	75
File: <i>lwarp.css</i> : Minor fixes for		File: <i>lwarp_tutorial.txt</i> added.	79
validation.	1		
File: <i>lwarpmk</i> used to compile		v0.22	
print, HTML, indexes, and		\LWR@parseDcolumn: Added tabular	
<i>lateximages</i>	1	D column.	377
Fix: \linewidth in a floatrow.	656	\LWR@parsebangcolumn: Added	
Moved sidebar and example code		tabular ! column.	372
to test suite.	1	\LWR@parsetablecols: Unknown	
Page geometry set to 6in wide with		table column types become 1.	
large margins.	180	Added tabular D, !, X columns.	379
Parallel versions of aux files for		\LWR@printmccoldata: Added	
print/HTML.	1	tabular D, !, and X columns.	401
Removed reliance on make, grep,		General: 2017/03/02	1
gawk.	1	abstract: Added.	548
Tabular: \unskip extra spaces.	361	changepage: Added.	583
Test Suite: HTML meta		dcolumn: Added.	618
descriptions.	1	ftnright: Added.	667
verbatim: Added.	348	geometry: Nullified commands.	669
BlockClass: Added optional style.	287	indentfirst: Added.	696
Renamed from "blockclass".	287	layout: Added.	699
LWR@nestspan: Fix: Minipages inside		lscape: Added.	710
a span.	283	mcaption: Added.	717
v0.21		nameref: Added.	739
\LWR@LwarpStart: Changed		nextpage: Added.	741
<i>lateximages</i> to a .txt file.	328	parskip: Added.	760
\LWR@filestart: Skip title if not		showkeys: Added.	783
given.	326	sidecap: Added.	784
\LWR@newhtmlfile: Skip title if not		tabularx: Added.	804
given.	311	varioref: Supported.	117
\marginpar: Fixed source listing.	305	verse: Added.	849
\marginparBlock: Fixed source			
listing.	305	v0.23	
General: 2017/02/23	1	\LWR@parsetablecols: Fix for vert	
fontenc: Added.	661	bar column type.	379
fontspec: Added.	661	\LWR@printmccoldata: Fix for vert	
inputenc: Added.	697	bar column type.	401
		General: 2017/03/02	1

v0.24	
\LRW@htmlfileref: Fix: Index links	
while \tracinglwarp.	427
\hspace: Add: \hspace \fill	
converts to 2em	536
\hypertocfloat: List of floats	
responds to lofdepth,	
lotdepth.	450
General: 2017/03/15	1
floatrow: Support for subfig. . . .	653
subfig: Added.	796
tikz: For tikz v3.0.0 or later,	
auto-loads tikz babel library if	
necessary.	816
Docs: Filename underscore. . . .	99, 112
Fix for inline images.	816
No longer preloads subcaption;	
conflicted with subfig.	182
picture: Fix for inline images. . .	515
v0.25	
\LRW@loadnever: Added the ability	
to prevent conflicting packages. .	172
\addcontentsline: Handles	
theorems.	442
General: 2016/03/22	1
amsthm: Added.	561
ccaption: Prevented.	582
ellipsis: Added.	622
emptypage: Added.	623
framed: Added.	665
lips: Added.	704
mdframed: Help avoid	
hyphenation.	719
nththeorem: Added.	744
showidx: Added.	783
theorem: Added.	811
Basic \LaTeX theorems: improved	
css.	349
Docs: Adds credits for patched	
code.	1
Docs: Testing lwarp.	152
Fix: Allows \XeTeX and \LuaTeX to	
preload graphics and graphicx. .	172
v0.26	
General: 2017/03/31	1
lwarp.css: Improved responsive	
marginpar and marginblock. .	205
cutwin: Added.	617
endnotes: Added.	624
floatflt: Added.	651
footmisc: Added.	661
footnotehyper: Added.	664
footnote: Added.	662
marginfix: Added.	716
marginnote: Added.	716
mparhack: Added.	734
pagenote: Supported as-is. . . .	757
sidenotes: Added.	785
Docs: Improved \MiTeX install	
instructions.	73, 75
Dollar span avoided in a	
lateximage.	463
Footnotes now are \LaTeX boxes	
instead of pagenotes.	300
lateximage: Labels track page	
numbers of lateximages.	497
Print mode now uses a minipage	
of \linewidth.	497
picture: Fix for \makebox in	
picture.	515
v0.27	
\LRW@footnotetext: Fix for table	
footnote par tags.	301
General: 2017/04/04	1
lettrine: Added.	700
microtype: Fix with \XeTeX ,	
\LuaTeX	731
soul: Added.	792
ulem: Added.	846
Docs: Installing utilities for	
MACOS.	77
Docs: Limitations of saveboxes. .	112
Page geometry modified to reduce	
line overflow.	180
v0.28	
\@wrindex: Improved indexing. . . .	453
\HTMLAuthor: Added \HTMLauthor.	
(Renamed in v0.30.)	299
\LRW@LwarpEnd: If FormatEPUB or	
FormatWP, no bottom nav. . . .	332
\LRW@LwarpStart:	
FormatWordProcessor forces	
single-file output.	328
\LRW@filestart: Adds HTML meta	
author.	326
\LRW@forcenewpage: Forces new	
PDF page before major	
environments.	279
\LRW@htmlcomment: Breaks ligatures	
in HTML comments.	285

\LWR@includegraphicsb: Adapts to graphics syntax.	676	\HTMLAuthor: Renamed from \HTMLauthor.	299
\LWR@newhtmlfile: If FormatEPUB or FormatWP: skips headers, footers, nav.	311	\HTMLDescription: Renamed from \NewHTMLdescription.	299
\LWR@parsetablecols: Added L, C, R, J column types.	379	\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	297
\LWR@startref: Removed space. . .	429	\HTMLLanguage: Renamed from \MetaLanguage.	326
\chapter: If EPUB, prints footnotes before each section.	323	\HTMLPageBottom: Renamed from \SetPageBottom.	297
\hyperindexref: Improved indexing.	454	\HTMLPageTop: Renamed from \SetPageTop.	297
\textup: Fixed span class.	527	General: 2017/04/29	1
General: 2017/04/14	1	lwarp-newproject removed, and combined with lwarp	202
glossaries : Added.	670	lwarpmk : Add: <code>xdyfile</code> configuration option.	248
graphics : Added.	672	lwarpmk : Fix: xindy and texindy adjusted for pdflatex , xelatex and lualatex	248
tabularx : Fix for optional pos. . .	804	lwarpmk : Fix: xindy now used for print index generation with latexmk	248
tabulary : Added.	805	lwarpmk : language now used for both index and glossary generation.	248
lwarpmk : Add: <i>printglossary</i> and <i>htmlglossary</i> commands.	248	File: <code>lwarp_html.xdy</code> renamed to <code>lwarp.xdy</code>	244
Added boolean FormatEPUB.	195	Fix: *.css files only written in print mode.	205
Added boolean FormatWP.	195	Fix: <code>lwarp.xdy</code> only written in print mode.	244
Added boolean <code>HTMLDebugComments</code>	190	Fix: <code>lwarp_mathjax.txt</code> : Only written in print mode.	245
Added boolean <code>HTMLMarkFloats</code> , changed to <code>WPMarkFloats</code> as of v0.42.	195	Option <code>lwarpmklang</code> changed to <code>IndexLanguage</code>	167
Docs: Modfying lwarpmk and index processing.	152	Option <code>OSWindows</code> replaces macro <code>\warpOSWindows</code>	168
File: <code>lwarp_mathjax.txt</code> : Updated CDN repository.	245	Option <code>xdyFilename</code> added.	168
Forced oneside to maintain large right margin.	180	Option latexmk replaces macro <code>\UseLatexmk</code>	169
v0.29		Options <code>HomeHTMLFilename</code> and <code>HTMLFilename</code> replace macros <code>\HomeHTMLFilename</code> and <code>\HTMLFilename</code>	168
\LWR@includegraphicsb: Fix: Error when no optional arguments. . .	676	v0.31	
General: 2017/04/15	1	General: 2017/05/15	1
*.lwarpmkconf: Add: language option for config files.	204	keyfloat : Improved compatibility.	697
lwarpmk.conf: Add: language option for config files.	203		
lwarpmk : Add: language option for config files.	248		
Add: <code>lwarpmklang</code> option for lwarp	167		
Docs: Using a glossary	90		
v0.30			
\CSSFilename: Renamed from \NewCSS.	298		

v0.32		\LWR@htmlelementclassline:	
\RequirePackage: Fix: Ignores		Moved optional argument in	
blanks in package list.	185	front of mandatory.	286
General: 2016/06/09	1	\LWR@htmlspanclass: Moved	
glossaries : Prevent error with		optional argument in front of	
\glo@name not defined.	454	mandatory.	284
lwarpmk : Fix: <i>io.lines()</i>		\LWR@nullfonts: Improved font	
changed to <i>file:lines()</i> due		control.	529
to luatex changes.	248	\LWR@restoreorigformatting:	
v0.33		booktabs : Works inside	
\HTMLAuthor: Fix: Provides empty		lateximage.	456
default author if none given. . .	299	Improved font control.	456
\LWR@loadbefore: Fix: No		\LWR@subhtmlelementclass:	
\PackageError if already		Moved optional argument in	
loaded.	171	front of mandatory.	285
\LWR@parseatcolumn: Fix: Column		\LWR@tabledatacolumnntag:	
alignment with leftmost @. . . .	372	booktabs : Works inside	
\LWR@tabledatasinglecolumnntag:		lateximage.	414
Fix: Macros in tabular could		\fboxBlock: Added.	524
cause extra data cell.	386	\makebox: Fix: Handles paren arg. .	521
\LWR@vspace: Add: \vspace		General: 2017/08/08	1
nullified.	538	babel-french : Adds fixed-width	
\StartDefiningTabulars: Add:		HTML spaces to punctuation. . .	281
Avoids error: misplaced		balance : Added.	571
alignment tab character &. . . .	271	booktabs : Works inside	
General: 2017/07/10	1	lateximage.	418, 575
amsmath : Removed <i>fleqn</i>		boxedminipage2e : Added.	577
option.	182	boxedminipage : Prevented.	576
fancyhdr : Fix: Optional args for		crop : Added.	616
\lhead, etc.	638	enumerate : Added.	625
Add: Tabular at and bang columns		enumitem : Added, no longer	
now have their own HTML		required.	625
columns.	361	everyshi : Added.	631
cleveref: Fix: Loaded		fancybox : Added.	633
\AtEndPreamble.	512	fancyvrb : Added, no longer	
Fix: Incorrectly-inline math		required.	640
environments.	481	figcaps : Added.	646
New handling of & to localize		filecontents : Required. Patched for	
catcode changes.	361	morewrites	181
v0.34		floatpag : Added.	652
\@fnsymbol: Text symbols instead of		flushend : Added.	659
math.	340	fullpage : Added.	668
\InlineClass: Moved optional		hyperxmp : Added.	694
argument in front of mandatory. .	288	idxlayout : Added.	695
\LWR@htmldivclass: Moved		marginfit : Added.	715
optional argument in front of		mdframed : Improved <i>mdtheorem</i>	
mandatory.	286	patch.	725
\LWR@htmlelementclass: Moved		moreverb : Added.	732
optional argument in front of		paralist : Added.	758
mandatory.	286	pdfscape : Added.	761

pdfsync: Added.	761	Force HTML superscripts.	301
prelim2e: Added.	763	\LWR@nullfonts: Fix: Filenames	
rotfloat: Added.	771	while using MATHJAX.	529
savetrees: Added.	772	\LWR@restoreorigformatting:	
shadow: Added.	783	siunitx: Improved	
syntonly: Added.	803	super/subscripts in a	
titles: No longer required.	817	<code>lateximage</code>	456
titleref: Prevented.	820	\LWR@section: Improved spacing.	318
xmpinl: Added.	874	\LWR@stoppars: Extra HTML source	
Added.	866	space after paragraphs.	295
Docs: Horizontal space		\fbox: Fix: Uses <code>\fboxrule</code> and	
limitations.	1	<code>\fboxsep</code>	523
Docs: Misplaced alignment		\framebox: Fix: Handles width and	
character.	154	horiz position.	522
File: <code>lwarp_mathjax.txt</code> : Version		\makebox: Fix: Handles width and	
change.	245	horiz position.	521
File: <code>README.txt</code> : updated.	1	General: 2017/08/17	1
Fix: Added the <code>eqnarray</code>		babel-french: Adjustments for	
environments.	481	French variants, load order,	
Improved font control.	526	footnotes, ellipses.	281
Lists refactored to remove		footnote: Extra HTML source space	
enumitem requirement.	351	after paragraphs.	662
Verbatim refactored to remove		siunitx: Fix for babel-french	505
fancyvrb requirement.	346	siunitx: Improved symbol	
<code>lateximage</code> : Fix: <code>lateximage</code> with		support.	786
<code>minipage</code> , <code>\parbox</code> , <code>\makebox</code> ,		transparent: Added.	843
<code>\fbox</code> , <code>\framebox</code> , <code>\raisebox</code> ,		upref: Added.	848
<code>\scalebox</code> , <code>\reflectbox</code>	497	xcolor: Added <code>\fcolorboxBlock</code> ,	
BlockClass: Moved optional		<code>\colorboxBlock</code>	858
argument in front of mandatory.	287	xcolor: Fix: Background none in	
<code>fminipage</code> : Added.	524	print mode.	858
\LWR@nestspan: Fix: Minipages,		xcolor: Refactored	
BlocksClass, and lists inside a		<code>\LWR@colorstyle</code>	862
span.	283	xcolor: Uses <code>\fboxrule</code> and	
\LWR@tabular: booktabs : Works		<code>\fboxsep</code>	858
inside <code>lateximage</code>	419	xcolor: <code>\fcolorbox</code> etc. now work	
v0.35		inside <code>lateximage</code>	858
General: 2017/08/08	1	Docs: Reorganized: Special cases	
Fix: <code>\textbf</code> and related.	526	and limitations.	109
v0.36		Source: Improved formatting.	1
\LWR@HTMLsanitize: Fix for		<code>lateximage</code> : Footnotes appear in	
babel-french	493	regular text instead of the	
\LWR@HTMLsanitizeexpand: Fix for		<code>lateximage minipage</code>	497
babel-french	494	\LWR@tabular: Fix for babel-french	419
\LWR@closeparagraph: Extra HTML		v0.37	
source space after paragraphs.	292	\include: Maintains independent	
\LWR@currenttextcolor: Fix for		aux files for HTML.	187
<code>\rule</code> when xcolor not loaded.	533	General: 2017/08/19	1
\LWR@footnotetext: Extra HTML		\LaTeX accents: Added.	201
source space after paragraphs.	301		

babel-french : Adjustment for load order.	281	Docs: Enhanced <i>Supported</i> <i>Features</i> table.	63
color : Prevented.	614	Docs: Index, tocbibind	122
comment : Maintains independent cutfiles for print, HTML.	170	Docs: Starred sections.	119
siunitx : Improved symbol support.	786	v0.39	
textcomp : Improved support. . .	806	\@maketitle : titling version. . . .	828
lwarpmk : Removes additional HTML aux files.	248	Native \LaTeX version.	341
File handles reorganized.	187	Removed minipages.	341, 828
v0.38		Supports authblk with <code><div></code> s of class <code>oneauthor</code> instead of <code>tabular</code>	341, 828
\@secCNTformat : Added for appendix	317	\AddSubTitlePublished : Added. .	342
\ForceHTMLPage : Added.	315	\LWR@domulticolumn : Add: Optional <code>vpos</code> and <code># rows</code>	404
\ForceHTMLTOC : Added.	315	\LWR@restoreorigformatting : Appended with <code>\appto</code> instead of calling various macros.	456
\LWR@section : <code>\part*</code> starts a new HTML page, for appendix	318	\LWR@tabledatacolumnntag : Don't start a data cell if see \TabularMacro	414
Modified spacing, uses \numberline	318	\ResumeTabular : Added.	413
\numberline : Added trailing <code>\quad</code>	449	\TabularMacro : Added.	413
\part : Fix with article class.	323	\multicolumnrow : Added. . .	411, 738
General: 2017/08/27	1	\printauthor : Removed minipages. . . .	337
appendix : Added.	566	Supports authblk with <code><div></code> s of class <code>oneauthor</code> instead of <code>tabular</code>	337
arabicfront : Added.	566	\thanksmarkseries : Removed minipage footnotes.	829
caption2 : Prevented.	582	General: 2017/09/05	1
chappg : Added.	584	a4wide : Added.	547
color : Forces xcolor as well.	614	a4 : Added.	547
fix2col : Added.	647	a5comb : Added.	547
fncychap : Added.	659	addlines : Added.	554
grffile : Added.	684	anySize : Added.	565
metalog : Added.	728	authblk : Added.	570
nonumonpart : Added.	743	bigdelim : Added.	573
nopageno : Added.	743	bigstrut : Added.	573
pagenote : Option page disabled. . . .	757	ebook : Added.	622
realscripts : Added.	766	fullwidth : Added.	668
relsize : Added.	768	midpage : Added.	732
romanbarpagenumber : Added. . .	770	multirow : Add: New optional <code>vpos</code> argument.	736
romanbar : Added.	770	multirow : Add: Supports left/right border for bigdelim	736
scalefnt : Added.	772	multirow : Fix: Long text argument.	736
siunitx : Removed from lwarp core.	786	supertabular : Added.	802
textcomp : Removed from lwarp core.	806	textarea : Added.	805
tocbibind : Added.	831	titling : Improved compatibility. .	825
xltxtra : Added.	873		
lwarpmk : Added <i>print1</i> and <i>html1</i> actions.	248		
Added <code>\markboth</code> , <code>\sloppy</code> , etc. . . .	279		

titling: Removed extraneous center environments.	826	graphics: Restores <code>\includegraphics</code> and <code>\DeclareGraphicsExtensions</code> in a <code>lateximage</code>	672
typearea: Added.	846	graphicx: Moved out of the <code>lwarp</code> core.	684
xtabular: Added.	875	grffile: Directly supported.	684
zwpagelayout: Added.	878	midfloat: Added.	731
Docs: Reorganized tabular discussion.	131	multirow: Improved bigdelim borders.	736
Titlepage <code>\published</code> and <code>\subtitle</code> removed. <code>\AddSubtitlePublished</code> restores.	342	pnfnote: Added.	762
titlepage: Clear pending footnotes.	336	quotchap: Added.	764
Removed minipages.	336	sectsty: Added.	781
titlingpage: Clear pending footnotes.	825	stabular: Added.	795
v0.40		tabs: Added.	804
<code>\@chapcntformat:</code> Added for tocbibind , anonchap	317	textcomp: Additional symbols, improved XeLaTeX and LuaLaTeX support.	806
<code>\LWR@HTMLhline:</code> Added.	418	tocbibind: Improved for <code>\simplechapter</code>	831
<code>\LWR@includegraphicsb:</code> Add: Full <code>\graphicspath</code> support.	676	xltxtra: Fix for <code>\showhyphens</code> with XeLaTeX.	873
<code>\LWR@nullfonts:</code> Fix: Long arguments for expandable command.	529	No longer preloads xfrac	182
<code>\LWR@restoreorigformatting:</code> Improved \LaTeX logos inside a <code>lateximage</code>	456	v0.41	
Improved symbols inside a <code>lateximage</code>	456	<code>\LWR@addcmidruletrim:</code> Add: <code>\cmidrule</code> trims.	392
Nullified <code>\InlineClass</code> , etc. inside a <code>lateximage</code>	456	<code>\LWR@clearmidrules:</code> Add: <code>\cmidrule</code> trims.	389
<code>\LWR@tabledatacolumnntag:</code> Fix for bigdelim : <code>\ldelim</code> , <code>\rdelim</code>	414	<code>\LWR@closetabledatacell:</code> Add: Mute > for <code>\bottomrule</code>	366
<code>\chapter:</code> Added support for quotchap	323	Fix: At/bang column with <code>\multirow</code>	366
<code>\multicolumnrow:</code> Fix: Adapts to older multirow and xparse	411	Fix: Cancel < for <code>\multicolumn</code>	366
<code>\simplechapterdelim:</code> Added for tocbibind , anonchap	317	<code>\LWR@domulticolumn:</code> Add: <code>\cmidrule</code> trims.	404
<code>\underline:</code> Added.	532	Added vertical rules.	405
General: 2017/09/25	1	<code>\LWR@nullifyNoAutoSpacing:</code> Fix: <code>\NoAutoSpacing</code> in a tabular with babel-french	419
adjmulticol: Added.	553	<code>\LWR@parsebarcolumn:</code> Added vertical rules.	374
anonchap: Added.	565	<code>\LWR@printatbang:</code> Add: <code>\cmidrule</code> trims.	385
bigdelim: Improved documentation.	573	Add: Mute at and bang columns for <code>\bottomrule</code>	385
cuted: Added.	616	<code>\LWR@printbartag:</code> Added vertical rules.	385
dblfnote: Added.	618	<code>\LWR@subaddcmidruletrim:</code> Added.	392
fnpos: Added.	660	<code>\LWR@subcmidrule:</code> Add: <code>\cmidrule</code> trims.	390
graphics: Moved out of the <code>lwarp</code> core.	672		

\LWR@tabulatedatasinglecolumnntag:		\LWR@includegraphicsb: Fix:	
Add: \cmidrule trims.	386	Filename expansion.	676
Add: Mute < for \bottomrule. . .	386	If FormatWP, use explicit size. . .	678
\LWR@tabularfinishrow:		\LWR@rememberetag: Fix: Numbering	
Unfinished tabular rows		and naming AMS math	
automatically filled.	368	environments.	495
\mcolrowcell: Added for		\LWR@restoreorigformatting:	
\multicolumnrow cells.	417	Improved \ensuremath.	458
General: 2017/10/07	1	\LWR@subaddcmidruletrim: Opt if	
multirow : Add: \cmidrule trims. 736		no rule given.	392
Added vertical rules.	737	\LWR@subsingledollar: If	
Fix: < spec.	737	FormatWP print LaTeX	
Improved rules.	575	expression.	464
\LWR@tabular: Fix: \NoAutoSpacing		\LWR@tabulatedatasinglecolumnntag:	
in a tabular with babel-french . 420		If FormatWP add cell alignment. 387	
v0.42		\LaTeX: If FormatWP use explicit	
\@ensuredmath: Improved		style.	542
\ensuremath.	472	\TeX: If FormatWP use explicit style. 542	
\@textsubscript: Added.	532	\hspace: If FormatWP add \quads. . 537	
\@textsuperscript: Added.	532	\listoffigures: Added boolean	
\LWR@HTMLhline: If FormatWP force		WPMarkLOFT.	446
explicit border.	418	\listoftables: Added boolean	
\LWR@HTMLtextstyle: Added.	526	WPMarkLOFT.	446
\LWR@addformatwpalignment: If		\marginpar: If FormatWP emulate a	
FormatWP add explicit style for		wrapfig.	305
cell alignment.	394	\marginparBlock: If FormatWP	
\LWR@addrulewidth: If FormatWP		emulate a wrapfig.	305
force explicit border.	393	\rule: If FormatWP add \quads. . . 540	
\LWR@amsmathbody: Fix: Numbering		\tableofcontents: Added boolean	
and naming AMS math		WPMarkTOC.	446
environments.	496	\underline: If FormatWP, use	
\LWR@amsmathbodynumbered: Fix:		explicit styles for \underline,	
Numbering and naming AMS		etc.	532
math environments.	496	General: 2017/10/30	1
\LWR@doequation: If FormatWP print		\textbf and related: If FormatWP,	
LaTeX expression.	476	use explicit styles for \textsc,	
\LWR@domulticolumn: If FormatWP		etc.	526
add cell alignment.	405	algorithmicx : If FormatWP add	
\LWR@doubledollar: If FormatWP		\quads.	559
print LaTeX expression.	470	epigraph : If FormatWP add HTML	
Improved \ensuremath.	470	styles.	626
Improved line spacing with		fancybox : If FormatWP add HTML	
mathjax.	470	styles.	633
\LWR@floatbegin: If FormatWP add		floatflt : Added width.	652
a text frame.	436	includegraphics : Fix: Class key. . 676	
\LWR@floatend: If FormatWP add a		keyfloat : If FormatWP add explicit	
text frame.	437	HTML style.	698
\LWR@htmlmathlabel: If FormatWP		moreverb : Simplified formatting	
print LaTeX expression.	480	of listings.	732
		morewrites : Added.	734

multirow : If FormatWP add cell alignment.	737	v0.43	\LWR@domulticolumn : Fix for vertical rules.	405
overpic : Added.	756		Fix: Multicolumn trim. . . .	404, 405
realscripts : Fix for subscripts in a lateximage.	766		\LWR@maybeprintpendingfootnotes : Added FootnoteDepth.	304
sidenotes : If FormatWP add explicit HTML style.	785		\LWR@nullfonts : Fix: Nullify dollar inside filenames.	529
siunitx : Improved \ensuremath	787		\LWR@parsetablecols : Ignore spaces in col spec.	379
soul : If FormatWP, add explicit styles.	792		\LWR@printmccoltype : Added vertical rules.	400
textcomp : Improved \interrobangdown	806		\LWR@restoreorigformatting : Fix: \ref in math or lateximage. . .	456
wrapfig : If FormatWP add explicit HTML style.	856		\LWR@section : Fix: Expansion in comparison.	318
Added boolean WPMarkLOFT	196		Fix: Math in section name. . . .	320, 322
Added boolean WPMarkMath	196		Fix: Nullify fonts inside HTML comment.	319
Added boolean WPMarkMinipages	195		\TabularMacro : \newcommand instead of \relax to fix supertabular and xtab	413
Added boolean WPMarkTOC	196		\href : Made robust.	433
Added boolean WPTitleHeading	196		\nameref : Made robust.	431
Docs: Added support page.	2		\nolinkurl : Made robust.	433
Docs: Improper \prevdepth	154		\url : Made robust.	434
Docs: Reorganized math limitations	123		General: 2017/11/08	1
File: lwarp_mathjax.txt : Updated siunitx script.	245		breakurl : Added.	577
Fix: Numbering and naming AMS math environments.	494		hyperref : Made robust. . . .	689, 691, 693
If FormatWP force explicit border. . . .	575		hyperref : \Gauge added.	693
If FormatWP, shift section headings.	197		luatodonotes : Added.	713
tabbing: Added.	349		todonotes : Added.	841
lateximage: Fix: Numbering and naming AMS math environments.	498		LWR@currentautosec : Added. . .	318
center: If FormatWP use explicit text-align.	503		Added FootnoteDepth.	300
minipage: Added boolean WPMarkMinipages	520		Docs: HTML settings table. . . .	98
If FormatWP add a text frame. . . .	518		Docs: Reorganized HTML customization.	98
eqnarray : Fix: Numbering and naming AMS math environments.	483	v0.44	\@currentlabelname : Adjustment for koma-script	424
If FormatWP print LaTeX expression.	482		\HTMLTitle : Added.	298
LWR@BlockClassWP : Added to factor code.	288		\LWR@addformatwpalignment : Fix for multicolumn alignment if FormatWP.	394
LWR@figcaption : If FormatWP forces italic captions.	440		\LWR@backgroundcolor : Added. . .	862
			\LWR@filestart : Add \HTMLTitle . . .	327
			Fix \HTMLAuthor	326
			\LWR@listitem : Added list and trivlist	354

<code>\LWR@patchlists</code> : Added list and trivlist.	357	<code>tocstyle</code> : Added.	839
<code>\LWR@strresult</code> : Fix: <code>\providecommand</code>	364	<code>todonotes</code> : Improved.	841
<code>\LWR@textcurrentcolor</code> : <code>xcolor</code> : Added <code>\LWR@textcurrentcolor</code>	861	<code>todo</code> : Added.	840
<code>\addcontentsline</code> : Automatic <code>\LWR@newfloatanchor</code>	442	<code>typearea</code> : Added expert commands.	846
<code>\chapter</code> : Add preamble for <code>koma-script</code>	323	<code>watermark</code> : Added.	856
<code>\marginparBlock</code> : Added.	305	<code>xcolor</code> : Added <code>\LWR@currenttextcolorstyle</code>	861
<code>\nopagecolor</code> : <code>xcolor</code> : Fix for <code>\nopagecolor</code>	863	<code>xcolor</code> : Added <code>\LWR@findcurrenttextcolor</code>	861
<code>\part</code> : Add preamble for <code>koma-script</code>	323	<code>xtab</code> : Fix for caption.	876
<code>\title</code> : Added <code>\thetitle</code>	298	Adjustment for <code>koma-script</code>	178
General: 2017/11/22	1	AMS environments: Fix: Groups for lateximages.	484
<code>algorithmicx</code> : Improved comment symbol.	559	If pdfLaTeX, require T1 and UTF8 encoding.	163
<code>atbegshi</code> : Added.	569	picture: <code>overpic</code> : Fix: Groups for lateximages.	515
<code>cancel</code> : Added.	578	list: Added list and trivlist.	354
<code>change page</code> : Additional options.	584	<code>LWR@nestspan</code> : Added list and trivlist.	283
<code>easy-todo</code> : Added.	621	v0.45	
<code>fancyref</code> : Added.	639	<code>\@currentHref</code> : Added.	432
<code>fixmetodonotes</code> : Added.	649	<code>\@donoparitem</code> : Modified for HTML.	352
<code>fixme</code> : Added.	648	<code>\@item</code> : Modified for HTML.	352
<code>fontenc</code> : Allowed after <code>lwarp</code>	661	<code>\@mklab</code> : Modified for HTML.	351
<code>hang</code> : Added.	685	<code>\CSSFilename</code> : Improved filenames with underscores.	298
<code>ifoddp page</code> : Added.	696	<code>\LWR@LwarpStart</code> : Fix: Lateximages on incorrect pages with Mathjax.	330
<code>ltxtable</code> : Added.	712	<code>\LWR@filenamenoblanks</code> : Fix: Section names with <code>\</code>	308
<code>luatodonotes</code> : Improved.	713	Fix: Section names with underscores.	308
<code>lwarp-patch-komascript</code> : Added.	879	<code>\LWR@includegraphicsb</code> : Improved URLs with underscores.	676
<code>overpic</code> : Fix: Groups for lateximages.	756	<code>\LWR@newautoidanchor</code> : Fix: No anchor if frozen autoid.	438
<code>pdfsync</code> : Fixes.	761	<code>\LWR@notmemoirloadafter</code> : Added.	171
<code>preview</code> : Added.	763	<code>\LWR@printpendingmpfootnotes</code> : Added.	304
<code>scrextend</code> : Added.	773	<code>\LWR@startref</code> : Fix: Labels with underscores.	430
<code>scrhack</code> : Added.	776	<code>\LWR@subhyperref</code> : Improved URLs with underscores.	432
<code>scr layer-notecolumn</code> : Added.	778	<code>\LWR@subhyperrefclass</code> : Improved URLs with underscores.	432
<code>scr layer-scr page</code> : Added.	778	<code>\LWR@sublabel</code> : Fix: Labels with underscores.	427
<code>scr layer</code> : Added.	776		
<code>section</code> : Added.	779		
<code>soulpos</code> : Added.	794		
<code>soulutf8</code> : Added.	794		
<code>supertabular</code> : Fix for caption.	802		
<code>tikz</code> : Fix: Groups for lateximages.	816		
<code>tocbasic</code> : Added.	830		
<code>tocloft</code> : Added <code>\newlistentry</code>	838		
<code>tocloft</code> : Improved <code>\newlistof</code>	839		

<code>\LWR@tabledatacolumnntag</code> : Fix	Docs: Improved MiKTeX install
Empty line between rows.	instructions.
416	73
<code>\chapter</code> : Add optional heading title	Docs: Moved table so doesn't
for memoir	interfere with install docs.
323	72
<code>\newpage</code> : Added.	File: <code>lwarp_mathjax.txt</code> : Allow
534	MATHJAX inside tabbing.
<code>\nolinkurl</code> : Fix: Underscore in URL.	File: <code>lwarp_mathjax.txt</code> : Allow
433	MATHJAX inside verse.
<code>\normalmarginpar</code> : Added.	Fix: Empty sidetoc.
306	445
<code>\reversemarginpar</code> : Added.	Improved: Robust <code>\</code> , <code>\</code> , and
306	<code>\textellipsis</code> commands.
<code>\section</code> : Add optional heading title	Separate <code>LWR@thisautoidWP</code> for
for memoir	word processor <code><div>s</code>
324	438
<code>\tableofcontents</code> : Fix: Empty	<code>thebibliography</code> : Patched to
sidetoc.	emphasize titles.
446	455
Fix: Patch <code>\AtBeginDocument</code>	<code>minipage</code> : Fix: Improper
446	<code>\prevdepth</code>
<code>\url</code> : Improved URLs with	518, 520
underscores.	v0.46
434	<code>\LWR@closeparagraph</code> : Fix: Tabular
General: 2018/01/14	empty lines.
1	293
array : Added.	<code>\LWR@closeprevious</code> : Fix: Stack
567	unnesting.
babel-french : Robust commands.	<code>\LWR@forcenewpage</code> : Fix: Improper
281	<code>\prevdepth</code>
backref : Added.	<code>\LWR@lookforpackagename</code> : Fix:
571	Spaces in <code>\usepackage</code>
breakurl : Fix: Underscore in URL.	<code>\LWR@providelength</code> : Added.
577	164
changebar : Added.	<code>\LWRPrintStack</code> : Name changed
583	from <code>\PrintStack</code>
cite : Added.	<code>\makebox</code> : Fix: Lateximage in a
613	<code>\makebox</code>
continue : Added.	<code>\popclose</code> : Fix: Stack unnesting.
616	269
endfloat : Added.	<code>\pushclose</code> : Fix: Stack unnesting.
623	268
epigraph : Support for memoir	General: 2018/01/23
627	1
fancyvrb : Improvements.	<code>LWR@tabularpardepth</code> added.
640, 641	364
flafter : Added.	amsthm : Adapted to <code>trivlist</code>
649	changes.
fltrace : Added.	563
658	mdframed : Fixes for svg math or
footnpag : Added.	<code>lateximage</code> in title.
664	721
fwlw : Added.	mdframed : Fixes for footnotes.
668	722
hanging : Added.	ntheorem : Adapted to <code>trivlist</code>
686	changes.
hyperref : Fix: Underscore in URL.	744
689	theorem : Adapt to <code>trivlist</code>
lwarp-patch-memoir : Added.	changes.
882	812, 814
memhfixc : Added.	<code>list</code> : Fix: Stack unnesting.
727	355
memoir : Added.	<code>LWR@tabular</code> : Fix: Tabular empty
545	lines.
natbib : Added.	422
739	v0.47
pagesel : Added.	<code>\LWR@LwarpStart</code> : Fix for svg math
758	in <code>\nameref</code>
prettyref : Added.	331
801	
subfigure : Added.	
801	
subfig : Fix for subcaption end tag.	
799	
subfig : Fix: Math in subcaptions.	
797	
textfit : Added.	
809	
titleref : Added.	
820	
turnthepage : Added.	
845	
Allows memoir 's preloaded	
packages.	
172	
Docs: xparse warnings.	
141	
Docs: Fix for double hyphens.	
77	
Docs: Improved install	
instructions.	
77	

<code>\LWR@WPcell</code> : Fix: Line wrap at HTML hyphen.	394	<code>keyfloat</code> : Fix for svg math in captions.	698
<code>\LWR@caption@begin</code> : Fix: Argument passed to <code>\LWR@origcaption@begin</code> . . .	441	<code>midpage</code> : Fix: Line wrap at HTML hyphen.	732
<code>\LWR@createautosec</code> : Fix: Line wrap at HTML hyphen.	316	<code>multirrow</code> : Fix: Line wrap at HTML hyphen.	736
<code>\LWR@domulticolumn</code> : Fix: Line wrap at HTML hyphen.	405	<code>multitoc</code> : Added.	739
<code>\LWR@float@begin</code> : Fix: Line wrap at HTML hyphen.	436	<code>ntheorem</code> : Fix: Line wrap at HTML hyphen.	748
<code>\LWR@html@closecomment</code> : Add <code>\mbox</code> to prevent line breaks. .	284	<code>realscripts</code> : Fix: Line wrap at HTML hyphen.	766
<code>\LWR@newautoidanchor</code> : Fix: Line wrap at HTML hyphen.	438	<code>scrextend</code> : Fix: Line wrap at HTML hyphen.	773
<code>\LWR@printopenlist</code> : Fix: Line wrap at HTML hyphen.	351	<code>sectionbreak</code> : Added.	780
<code>\LWR@startref</code> : Fix: Line wrap at HTML hyphen.	429	<code>sidenotes</code> : Fix for svg math in captions.	785
<code>\LWR@sublabel</code> : Fix: Line wrap at HTML hyphen.	428	<code>subfig</code> : Fix for svg math in captions.	797
<code>\LWR@subsingledollar</code> : Added svg math image baseline adjust and em sizing.	464	<code>subfig</code> : Fix: Support <code>\nameref</code> . . .	796
Fix: Line wrap at HTML hyphen. .	468	<code>xurl</code> : Added.	877
<code>\captionlistentry</code> : Fix: Line wrap at HTML hyphen.	442	<code>lwarpmk</code> : <code>pdfcrop</code> : Removed hires option for improved crop accuracy.	248
<code>\hypertoc</code> : Fix: Line wrap at HTML hyphen.	450	<code>lateximage</code> : Added css style option. .	497
<code>\hypertocfloat</code> : Fix: Line wrap at HTML hyphen.	451	Fix: Line wrap at HTML hyphen. .	501
<code>\makebox</code> : Fix: Line wrap at HTML hyphen.	522	<code>center</code> : Fix: Line wrap at HTML hyphen.	503
General: 2018/01/30	1	<code>minipage</code> : Fix: Line wrap at HTML hyphen.	518, 519
<code>adjmulticol</code> : Fix: Line wrap at HTML hyphen.	553	<code>flushleft</code> : Fix: Line wrap at HTML hyphen.	503
<code>blowup</code> : Added.	574	<code>flushright</code> : Fix: Line wrap at HTML hyphen.	503
<code>caption</code> : Added.	579	<code>enumerate</code> : Fix: Line wrap at HTML hyphen.	356
<code>caption</code> : Also loads <code>lwrap-caption</code>	186	<code>itemize</code> : Fix: Line wrap at HTML hyphen.	355
<code>change page</code> : Fix for pagecheck macros.	584	<code>LWR@BlockClassWP</code> : Fix: Line wrap at HTML hyphen.	288
<code>endheads</code> : Added.	623	<code>v0.48</code>	
<code>epigraph</code> : Fix: Line wrap at HTML hyphen.	626	<code>\@@@setcpageref</code> : Fix for new v0.21 of <code>cleveref</code>	514
<code>hanging</code> : Fix: Line wrap at HTML hyphen.	687	<code>\@@@setcref</code> : Fix for new v0.21 of <code>cleveref</code>	513
<code>hang</code> : Fix: Line wrap at HTML hyphen.	685	<code>\@@@setcrefrange</code> : Fix for new v0.21 of <code>cleveref</code>	513
		<code>\@biblabel</code> : Improved bibliography label.	455
		<code>\@item</code> : Honors <code>\makelabel</code>	352

\@maketitle: Fix: Errors with	General: 2018/02/14	1
IEEEtran class.	acronym : Added.	552
\LWR@LwarpStart: Adjusted space	acro : Added.	550
around captions.	chapterbib : Added.	584
\LWR@ProvidesPackageDrop: Fix:	colortbl : Added.	382, 395, 614
Options with braces.	fancyheadings : Prevented.	638
\LWR@addtabularhrulecolor:	fancyref : Now directly supported.	639
colortbl : Added.	hyppcap : Added.	687
\LWR@addtabularrulecolors:	hypernat : Added.	688
colortbl : Added.	hyperref : \texorpdfstring now	
\LWR@closetabledatcell:	uses the \TeX string.	692
colortbl : Added.	luatodonotes : Improved	
\LWR@includegraphicsb: Fix:	\todotoc.	713
Virtual page size limited to a	siunitx : Changes fraction to	
group.	symbol.	789
\LWR@lookforpackagename: Fix:	siunitx : Improved svg math.	787, 788
Parsing similar package names.	siunitx : Improved color output.	787
\LWR@newautopagelabel: Fix: TOC,	stfloats : Added.	795
LOF, LOT links.	todonotes : Improved	
\LWR@newhtmlfile: Fix: TOC, LOF,	\todotoc.	841
LOT links.	vmargin : Added.	852
\LWR@nullfonts: Fix: \newline in	xfrac : Fix: Added groups around	
title.	super/subscripts to localize	
\LWR@parsedrequirepackagenames:	LWR@nestspan changes.	871
Fix: Parsing similar package	Docs: Converting an existing	
names.	document.	93
\LWR@parsetablecols: Fix: Ignore	Improved font control.	530
optional tabular column	lateximage : Print mode boxed to	
arguments.	natural width.	502
\LWR@restoreorigformatting: Fix:	abstract : Allow optional name.	343
Font sizes in svg math,	LWR@tabular: colortbl : Added.	420
lateximage , Tikz	v0.49	
Fix: Spacing in svg math,	\LWR@addtabularcellcolor:	
lateximage , Tikz	xcolor : Added tabular row colors.	398
\LWR@section: Fix: TOC, LOF, LOT	\LWR@domulticolumn: xcolor : Added	
links.	tabular row colors.	405
\LWR@tabledatasinglecolumn:tag:	\LWR@printlength: Fix: Group	
colortbl : Added.	printlen changes.	182
\LWR@textcurrentfont: Added.	affiliation : Fix: Adapts to classes	
Improves font control.	which already provide.	335
\bibliography: Fix: \BaseJobname	href : Fix: Adapt to classes.	433
for bibliography.	noalign : Fix: \noalign inside	
\centerline: Added.	tabular.	418
\l@part: Adapts to classes without	url : Fix: Adapt to classes.	434
\part.	General: 2018/02/19	1
\leftline: Added.	amsmath : Fix: Patches for	
\mbox: Nullified for HTML.	\eqref.	182
\rightline: Added.	eso-pic : Fix for	
\thempfootnote: Removed	\AddToShipoutPicture.	630
\itshape.	figsize : Added.	647

fnlineno : Added.	660	lwarp_one_limage.txt : Added.	245
hypdestopt : Added.	688	amsmath : Fix: Upright tags for svgmath.	182
hyphenat : Added.	694	axodraw2 : Added.	571
lineno : Added.	701	bytefield : Added.	578
luacolor : Added.	712	dblfloatfix : Added.	618
pagegrid : Added.	757	diagbox : Added.	619
pdfrender : Added.	761	epstopdf : Added.	627
resizegather : Added.	769	listings : Force flexible columns.	704
vertbars : Added.	851	morefloats : Added.	732
vwcol : Added.	853	nonfloat : Added.	742
xcolor : Added tabular row colors.	382, 867	ntheorem : Fix: Not standard nor amsthm selected.	751
Fix: Adapt to classes.	534	pbox : Added.	760
v0.50		phfqit : Added.	762
\@ensuredmath: Fix: Use lateximage even if MathJax.	472	schemata : Added.	772
Improved svg math alt tags.	472	siunitx : Fix: Loads xcolor	786
\LWR@atbeginverbatim: Improved column alignment.	347	siunitx : Improved svg math alt tags.	788
\LWR@doequation: Improved svg math display.	477	siunitx : Improved units.	505, 787, 789
\LWR@doubledollar: Improved svg math alt tags.	470	xy : Added.	877
Improved svg math display.	470	lwarpmk : Error if lateximages.txt does not exist.	248
\LWR@footnotetext: Robustify macros.	302	lwarpmk : Error if lwarpmk.conf points to lwarp	248
\LWR@htmlrefsectionfilename: Fix: SVG math in a section name.	276	lwarpmk : Improved error messages.	248
\LWR@newhtmlfile: Fix: SVG math in a section name.	312	lwarpmk : MD5 hash avoids duplicate svg math.	248
\LWR@nullfonts: Fix: \underline in sectioning file name.	529	lwarpmk : Multiprocess support making lateximages.	248
Robustify macros.	529	AMS environments: Improved svg math display.	484
\LWR@overline: Added.	533	Fix: \centering, etc. in svg math, lateximage, Tikz.	200
\LWR@restoreorigformatting: Fix: \centering, etc. in svg math, lateximage, Tikz.	457	Fix: Load fontspec if necessary.	179
\LWR@subsingledollar: Fix: Use lateximage even if MathJax.	464	Robustify macros.	530
Improved svg math alt tags.	464	lateximage: Fix: SVG math in a section name.	500
MD5 hash avoids duplicate svg math.	468	MD5 hash avoids duplicate svg math.	498, 501
\LWR@vspace: Robustify macros.	538	eqnarray : Improved svg math display.	483
\newline: Robustify macros.	534	v0.51	
\textsubscript: Robustify macros.	532	\@ensuredmath: Hashes \ensuremath.	472
\textsuperscript: Robustify macros.	532	\@item: Restored list label space.	353
General: 2018/03/03	1	\LWR@HTMLsanitize: Fix: Escapes double quotes.	493
lwarp.css: Improved svg display math centering.	205		

<code>\LWR@HTMLsanitizeexpand</code> : Fix: Escapes double quotes.	494	<code>lwarp_one_limage.txt</code> :	
<code>\LWR@LwarpStart</code> : MathJax: Nullifies <code>\ensuremath</code>	331	<code>pdftocairo -noshrink</code> added. . .	245
<code>\LWR@addbaselinemarker</code> : Improved svg math baseline. . .	463	<code>afterpackage</code> : No longer required. .	180
<code>\LWR@atbeginverbatim</code> : Adds vertical offset.	347	<code>chemfig</code> : Added.	585
<code>\LWR@customizeMathJax</code> : MathJax: Nullifies <code>\ensuremath</code>	310	<code>chemformula</code> : Added.	586
<code>\LWR@doequation</code> : Fix: <code>\addcontentsline</code> inside svg math. Provides an autoid anchor. .	477	<code>chemgreek</code> : Added.	591
<code>\LWR@doubledollar</code> : Fix: <code>\addcontentsline</code> inside svg math. Provides an autoid anchor. .	470	<code>chemmacros</code> : Added.	592
<code>\LWR@findcurrenttextcolor</code> : Added <code>\LWR@findcurrenttextcolor</code> when no <code>xcolor</code>	533	<code>chemnum</code> : Added.	612
<code>\LWR@newautoidanchor</code> : Fix: No autoid is inside a <code>lateximage</code> . .	438	<code>epstopdf-base</code> : Added.	628
<code>\LWR@newhtmlfile</code> : MathJax: Nullifies <code>\ensuremath</code>	314	<code>fancybox</code> : Fix: Optional tag for <code>\item</code> in a span.	636
<code>\LWR@subsingledollar</code> : Fix: <code>\ensuredmath</code> inside svg image. .	465	<code>grid</code> : Added.	684
Fix: <code>lateximage</code> inside \mathcal{MS} <code>\text</code>	466	<code>listings</code> : Forces cleared options. .	705
Fix: Honors text font around svg math.	465	<code>ltxgrid</code> : Added.	711
Fix: SVG math with enclosed <code>lateximage</code>	465	<code>mhchem</code> : Added.	729
Improved svg math baseline. . . .	466	<code>tikz</code> : Fix for <code>\tikz</code> macro.	816
SVG math baseline improved with invisible rule at corner.	469	<code>tikz</code> : Fix for <code>tikz</code> with optional argument.	816
Typeset svg math only once during measurement.	466	<code>titling</code> : Fix for <code>\thanks</code> mark. . . .	827
<code>\LWR@textcurrentcolor</code> : <code>xcolor</code> : <code>\LWR@textcurrentcolor</code> if <code>xcolor</code> not loaded.	533	<code>lwarpmk</code> : <code>pdfcrop</code> : Restored hires option.	248
<code>\addcontentsline</code> : Add missing support for float mechanism if necessary.	443	<code>lwarpmk</code> : <code>pdftocairo -noshrink</code> added.	248
No anchor ID if inside svg image. .	443	AMS environments: Fix: <code>\addcontentsline</code> inside svg math. Provides an autoid anchor. .	484
<code>\displaymathnormal</code> : Processing for complicated display math. . . .	479	Docs: <code>tikz</code> limitations.	130
<code>\displaymathother</code> : Processing for complicated display math. . . .	479	Docs: Multiple authors and affiliations.	119
<code>\textcolor</code> : Fix: SVG math color. . .	863	Docs: Things to avoid.	110
General: 2018/03/24	1	Docs: Updated Converting an existing document.	93
		Fix: Remember original <code>\#</code> in case is redefined.	198
		HTML entity used for text dollar. .	463
		<code>lateximage</code> : Added additional hashing option.	497
		Fix: <code>lateximage</code> inside \mathcal{MS} <code>\text</code>	497
		Processing for complicated display math.	500
		<code>alignat</code> : Fix: Added.	489
		<code>eqnarray</code> : Fix: <code>\addcontentsline</code> inside svg math. Provides an autoid anchor.	483
		<code>LWR@displaymathother</code> : Processing for complicated display math. . .	473
		<code>LWR@equationother</code> : Processing for complicated display math. . . .	473

v0.52	
\@ensuredmath: Improved hashing expansion.	472
\@mpfootnotetext: Fix: Paragraph handling.	303
\CustomizeMathJax: Added.	310
\LWR@addbaselinemarker: Warnings if	
lwrap_baseline_marker.png is not present or if graphicx/s not loaded.	463
\LWR@customizedMathJax: Added.	310
\LWR@doequation: Fix: equation* now based on equation instead of displaymath.	477
Fix: equation* with split.	476
\LWR@filenamenoblanks: Fix: \FileDepth with non-utf8 encoding.	309
\LWR@footnotetext: Fix: Paragraph handling.	302
\LWR@newhtmlfile: Fix: \FileDepth with non-utf8 encoding.	313
\LWR@nullfonts: Fix: \texorpdfstring in section names.	530
\LWR@section: Fix: Footnote numbering: Limited HTML comment if starred.	319
Fix: Footnote numbering: Use short TOC entry for HTMLDebug comments.	319
\LWR@subsingledollar: Added user-adjustable svg math font scaling.	467
\LateximageFontScale: Added user-adjustable svg math font scaling.	492
\href: Fix: #, %, &, ~, _ in URL.	433
\nolinkurl: Fix: #, %, &, ~, _ in URL.	433
\url: Fix: #, %, &, ~, _ in URL.	434
General: 2018/04/01	1
breakurl: Fix: #, %, &, ~, _ in URL.	577
endfloat: Updated for v2.6.	623
fancyvrb: Initial support for \VerbatimFootnotes.	633, 640
hyperref: Fix: #, %, &, ~, _ in URL.	689, 691
nicefrac: Added.	742
url: Added.	849
lwrapmk: Fix: Memory overflow when spawning tasks.	248
lwrapmk: Fix: Skip image generation if from page 0.	248
Changed FootnoteDepth default to \subsubsection.	300
Docs: Improved install instructions.	74
Fix: MathJax script line wraps. Reduced right margin.	180
If pdfLaTeX, allow other input encoding.	163
pkggraphics: Added defaults.	674
pkggraphicx: Updated for v1.1a.	675
pkggraphicx: Updated for v1.1b.	675
Restore \kill in a lateximage.	710
tabbing: Fix to allow inside lateximage.	349
lateximage: Fix for hash expansion.	499
v0.53	
General: 2018/04/01	1
lwrapmk: Added	
lwrapmk cleanimages.	248
lwrapmk: Added warning for corrupted images.	248
Docs: lwrapmk cleanimages.	91
Docs: lwrapmk pdftohtml.	91
v0.54	
\LWR@afterendverbatim: Added vspace argument.	348
\LWR@atbeginverbatim: Improved column alignment.	347
\LWR@earlyloadnever: Added.	172
\LWR@endfloatalignment: Honor \centering, etc. in floats.	439
\LWR@floatalignment: Honor \centering, etc. in floats.	439
\LWR@floatbegin: Honor \centering, etc. in floats.	437
\LWR@floatend: Honor \centering, etc. in floats.	437
\LateximageFontSizeName: Defaults to normalsize.	492
\centering: Added debug comment.	503
\raggedleft: Added debug comment.	503
\raggedright: Added debug comment.	504

General: 2018/04/22	1	lwarpmk : Changed language to xindylanguage.	248
*.lwarpmkconf: Option IndexLanguage changed to xindyLanguage.	204	lwarpmk : Changed xdyfile to xindystyle.	248
*.lwarpmkconf: Option pdftotextEnc added.	204	lwarpmk : Improved error if configuration file does not exist.	248
*.lwarpmkconf: Option xdyFilename changed to xindyStyle.	204	lwarpmk : Increased prominence for error for an unknown command.	248
*.lwarpmkconf: Option xindyCodepage added.	204	lwarpmk : Verifies HTML version exists before lwarpmk limages.	248
lwarp.css: Fix: Text-decoration-skip: auto.	205	lwarpmk : Verifies image references before lwarpmk limages.	248
lwarpmk.conf: Option IndexLanguage changed to xindyLanguage.	203	Add: pdftotextEnc.	168
lwarpmk.conf: Option pdftotextEnc added.	203	Add: xindyCodepage.	168
lwarpmk.conf: Option xdyFilename changed to xindyStyle.	203	Added early check for disallowed packages.	172
lwarpmk.conf: Option xindyCodepage added.	203	Docs: BibTeX.	121
aecc : Prevented.	554	Docs: Macros in sectioning names.	110
ae : Prevented.	554	Option IndexLanguage changed to xindyLanguage.	167
bibunits : Added.	572	Option xdyFilename changed to xindyStyle.	168
chnpage : Added.	584	verse: Fix: Line spacing.	345
forest : Added.	664	v0.55	
glossaries : Fix when not using babel or polyglossia.	671	\LWR@LwarpStart: Fix: Overfull boxes in lateximages.	329
gridset : Added.	684	\LWR@floatbegin: Fix: Float optional args.	437
hyperref : Fix: \hyperref and \hyperlink with special chars in text.	691	\LWR@nullfonts: Removed extraneous space which appeared in file links.	530
hyperref : Fix: \ref in \hyperref and \hyperlink caused nested link.	691	\phantomsection: Fix: \ForceHTMLTOC with \phantomsection.	541
inputenx : Added.	697	General: 2018/04/26	1
magaz : Added.	715	clrdblpg : Added.	613
memoir : Update for v3.7g.	887	Fix: \centering, etc. for koma-script.	437
ragged2e : Fix: \centering, etc.	766	Fix: QED symbols in lateximage.	564, 756
tlenc : Prevented.	803	v0.56	
textcomp : Fix for \textperthousand.	806	\LWR@addcdashline: arydashln: Added.	394
tikz : Fixes for pgfpicture, minipages, fit, align, font.	816	\LWR@addmulticolvertrulecolor: Adds support for dashed vertical rules.	403
wasysym : Prevented.	856		
lwarpmk : Added pdftotextenc.	248		
lwarpmk : Added xindycodepage.	248		

Adds support for double vertical rules.	403	lwarp.css: Added span.textbf, etc.	205
\LWR@addtabularhrulecolor: Adds support for arydshln dashed rules.	395	lwarpmk.conf: Records -shell-escape.	203
Adds support for double \hlines and \midrules.	395	arydshln : Added.	361, 567
\LWR@addtabularrulecolors: Adds support for dashed vertical rules.	396	lua-check-hyphen : Added.	712
Adds support for double vertical rules.	396	paralist : Fixes for compactenum, compactitem, compactdesc.	758
\LWR@closeparagraph: Added support for parnotes	293	parnotes : Added.	759
\LWR@domulticolumn: Adds support for dashed vertical rules.	405	quoting : Added.	765
Adds support for double vertical rules.	405	tocenter : Added.	832
\LWR@floatbegin: Adds a <class> per float package style.	436	underscore : Added.	848
\LWR@multicolpartext: Fix: \multicolumn parameters.	400	lwarpmk : Added lwarpmk pdftosvg.	248
\LWR@openparagraph: Added support for parnotes	292	lwarpmk : Supports -shell-escape.	248
\LWR@parsebarcolumn: Adds support for double vertical rules.	374	Added \thinspace.	533
\LWR@parsecoloncolumn: arydshln : Added.	375	Docs: <i>lwarpmk pdftosvg</i>	91
\LWR@parsesemicoloncolumn: arydshln : Added.	375	LWR@blocktextcurrentfont: Added div.textbf, etc.	530
\LWR@parsetablecols: Added array W column.	381	v0.57	
\LWR@printmccoldata: Added array W column.	401	\BlockClassSingle: Improved print/HTML output selection.	288
Fix: \multicolumn parameters.	401	\InlineClass: Improved print/HTML output selection.	288
\LWR@printmccoltype: Added array W column.	399	\LWR@customizeMathJax: MathJax: Supports \footnote, \footnotemark.	310
Adds support for dashed vertical rules.	400	\LWR@ref@ignorestar: subcaption : Fix: \subref.	430
Adds support for double vertical rules.	400	\LWR@subhyperref: Fix: Text catcodes.	432
\LWR@tabledatacolumnntag: Fix: \morecmidrules	415	\LWR@subhyperrefreftext: Fix: Text catcodes.	432
\LWR@textcurrentfont: Added span.textbf, etc.	530	\LWR@subsingledollar: Fix: Dynamic inline math expressions.	465, 466, 468
General: 2018/05/12	1	\LWR@vspace: Improved print/HTML output selection.	538
*.lwarpmkconf: Records -shell-escape.	204	\StartDefiningMath: Added.	271
lwarp.css: Added div.textbf, etc.	205	\StartDynamicMath: Added.	272
		\boxframe: xcolor : Fix: Colored \boxframe.	867
		\colorbox: xcolor : New system for switching print and HTML outputs.	864
		\colorboxBlock: xcolor : New system for switching print and HTML outputs.	864

<code>\fboxBlock</code> : Improved print/HTML output selection.	524	<code>algorithm2e</code> : Added.	555
<code>\fcolorbox</code> : xcolor : New system for switching print and HTML outputs.	864	<code>bigdelim</code> : Improved print/HTML output selection.	573
<code>\framebox</code> : Improved print/HTML output selection.	522	<code>breakurl</code> : Fix: Text catcodes. . . .	577
<code>\href</code> : Fix: Text catcodes.	433	<code>colortbl</code> : New system for switching print and HTML outputs.	614, 615
<code>\listof</code> : Fix: Provide <code>\l@name</code> if not defined.	447	<code>ellipsis</code> : Added <code>\midwordellipsis</code>	622
<code>\makebox</code> : Improved print/HTML output selection.	521	<code>errata</code> : Added.	628
<code>\mathimagename</code> : Added.	462	<code>float</code> : Added float styles.	650
<code>\mbox</code> : Improved print/HTML output selection.	521	<code>float</code> : Fix: Do not pre-define <code>\l@name</code>	651
<code>\multicolumnrow</code> : multirow : Improved print/HTML output selection.	738	<code>glossary</code> : Prevented.	671
Improved print/HTML output selection.	411	<code>ltablex</code> : Added.	710
<code>\newfloat</code> : rotfloat : Added float styles.	771	<code>marginnote</code> : Fix: Long optional argument.	716
rotfloat : Fix for listof sideways floats.	772	<code>multirow</code> : Improved print/HTML output selection.	736
<code>\packagediagramname</code> : Added. . . .	462	<code>register</code> : Added.	767
<code>\parbox</code> : Improved print/HTML output selection.	521	<code>subcaption</code> : Fix: <code>\subref</code>	689
<code>\raisebox</code> : Improved print/HTML output selection.	526	<code>trimclip</code> : Added.	843
<code>\reflectbox</code> : Improved print/HTML output selection.	683	<code>vowel</code> : Added.	852
<code>\resizebox</code> : Improved print/HTML output selection.	683	<code>xellipsis</code> : Added.	869
<code>\rotatebox</code> : Improved print/HTML output selection.	681	<code>xfrac</code> : Improved print/HTML <code>\scalebox</code> control.	871
<code>\rule</code> : Fix: Colored rules.	540	<code>xtabular</code> : Added.	873
<code>\scalebox</code> : Improved print/HTML output selection.	682	<code>xpiano</code> : Added.	874
<code>\textcolor</code> : xcolor : New system for switching print and HTML outputs.	863	<code>lwarpmk</code> : Improved code factoring.	248
General: 2018/06/06	1	<code>lwarpmk</code> : Improved error handling.	248
<code>lwarp.css</code> : Added ruled, boxed, boxruled floats.	205	Docs: Recompiling <code>lwarpmk</code> or CSS files.	152
<code>lwarp.css</code> : Increased float vertical margins.	205	Docs: Recreating the index for <code>lwarp</code> source.	150
		New system for switching print and HTML outputs.	193
		<code>minipage</code> : Improved print/HTML output selection.	517
		<code>BlockClass</code> : Improved print/HTML output selection.	287
		<code>fminipage</code> : Improved print/HTML output selection.	524
		<code>LWR@BlockClassWP</code> : Improved print/HTML output selection. . . .	288

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	
\\$	<u>462</u>
\&	<i>274</i> , <u>6276</u>
\(<u>8644</u>
\)	<u>8644</u>
\,	<i>111</i>
-shell-escape (option)	<i>94</i>
\@@@setcpageref	<u>9671</u>
\@@@setcref	<u>9634</u>
\@@@setcrefrange	<u>9648</u>
\@author	<i>334</i>
\@begintheorem	<u>5942</u>
\@biblabel	<u>8287</u>
\@capttype	<u>7934</u>
\@chapcntformat	<u>5234</u>
\@currentHref	<u>7782</u>
\@currentlabelname	<u>7640</u>
\@date	<i>334</i>
\@dlbfloat	<u>7900</u>
\@donoparitem	<u>5968</u>
\@endtheorem	<u>5954</u>
\@ensuredmath	<u>8651</u>
\@float	<u>7900</u>
\@fnsymbol	<u>5725</u>
\@footnotetext	<u>4885</u>
\@include	<u>671</u>
\@item	<u>5981</u>
\@makecaption	<u>7955</u>
\@makefnmark	<u>4853</u>
\@makefntext	<u>4852</u>
\@maketitle	<u>56</u> , <u>5748</u>
\@mklab	<u>5962</u>
\@mpfootnotetext	<u>4887</u>
\@nbitem	<u>6048</u>
\@opargbegintheorem	<u>5948</u>
\@rowc@lors	<u>6591</u>
\@rowcolors	<u>6590</u>
\@seccntformat	<u>5232</u>
\@starttoc	<u>8078</u>
\@textsubscript	<u>10126</u>
\@textsuperscript	<u>10122</u>
\@title	<i>334</i>
\@wrglossary	<u>8262</u>
\@wrindex	<u>8253</u>
\\	<i>535</i>
\$	<i>470</i>
\$\$	<i>470</i>
~	<i>111</i>
A	
a4 (package)	<i>547</i>
a4wide (package)	<i>547</i>
a5comb (package)	<i>547</i>
abstract (environment)	<u>5811</u>
abstract (package)	<i>120</i> , <i>547</i>
\abstractname	<i>101</i> , <u>5810</u>
accents	<i>314</i>
acro (package)	<i>550</i>
acronym (package)	<i>551</i>
adapting	
class	<i>152</i>
document	<i>93</i>
package	<i>150</i>
\addcontentsline	<u>8029</u>
addlines (package)	<i>554</i>
\AddSubtitlePublished	<u>5776</u>
adjmulticol (package)	<i>553</i>
Adobe (program)	<i>69</i>
ae (package)	<i>554</i>
aecc (package)	<i>554</i>
affiliation	
multiple authors	<i>119</i>
\affiliation	<u>5628</u>
afterpage (package)	<i>555</i>
algorithm2e (package)	<i>555</i>
algorithmic	
with newfloat, trivfloat	<i>845</i>
algorithmicx (package)	<i>559</i>
algorithmx (package)	<i>136</i>
align (environment)	<u>9050</u>
align* (environment)	<u>9076</u>
alignat (environment)	<u>9154</u>
alignat* (environment)	<u>9180</u>
alignment	
figure or table	<i>136</i>

alignment tab character &, misplaced	132, 270, 358	bigdelim (package)	135, 572
alltt (package)	560	bigstrut (package)	573
alt tags	127	BlockClass (environment)	4579
\AmS	10364	\BlockClassSingle	4593
AMSMath		blowup (package)	574
split miss-numbered	128, 744	bookmark (package)	574
amsmath (package)	182	booktabs (package)	575
amsthm (package)	561	boolean:	
\and	333	CombineHigherDepths	100, 307
anonchap (package)	565	FileSectionNames	100, 275
anysize (package)	565	FormatEpub	142, 195
appendix (package)	120, 566	FormatWP	144, 195
arabicfront (package)	566	HTMLDebugComments	190
array		LWR@amsmultline	480
chemformula	127	LWR@copiedsidetoc	445
mhchem	728	LWR@doingapar	290
array (package)	567	LWR@doingcmidrule	363
\arrayrulecolor	6598	LWR@doingstartpars	290
\arrayrulecolornexttoken	6598	LWR@doingtbrule	363
arydshln (package)	567	LWR@dynamicmath	272
AsciiDoc (program)	69	LWR@emptyatbang	363
AsciiDoctor (program)	69	LWR@exitingtabular	414
Asciidoctor-LaTeX (program)	69	LWR@freezethisautoid	438
atbegshi (package)	569	LWR@indisplaymathimage	462
\attrib	138, 345, 849	LWR@infloatrow	676
\attribution	5821	LWR@intabularmetadata	363
authblk (package)	120, 570	LWR@isstartingequation	495
author		LWR@mathmacro	271
HTML meta tag	109, 299	LWR@minipagefullwidth	517
multiple	119	LWR@minipagethispar	517
\author	108, 333	LWR@opttablecol	379
autosec	316	LWR@origmathjax	166
axodraw2 (package)	571	LWR@skipatbang	363
		LWR@skippingmcolrowcell	363
		LWR@skippingmrowcell	363
		LWR@starredlongtable	406
		LWR@startedrow	362
		LWR@tableparcell	363
		LWR@tabularcelladded	362
		LWR@tabularmutemods	414
		LWR@tracinglwarp	189
		LWR@validtablecol	377
		LWR@verbtags	346
		mathjax	166
		usingOSWindows	165
		warpingHTML	166
		warpingprint	166
		WPMarkFloats	146, 195
		WPMarkLOFT	147, 196
		WPMarkMath	147, 196
B			
babel (package)	139		
\backmatter	5213		
backref (package)	571		
balance (package)	571		
\BaseJobname	4245		
BaseJobname (option)	97, 167		
baseline			
tabular	378		
\bfseries	10111		
\bibliography	8278		
BibTeX			
\etalchar error	121		
\BibTeX	10359		
bibunits (package)	572		

WPMarkMinipages	146, 195	\cleardoublepage	10256
WPMarkTOC	146, 196	\clearpage	10256
WPTitleHeading	147, 196	cleveref (package)	117, 512
boxedminipage (package)	576	clrdblpg (package)	613
boxedminipage2e (package)	577	cmap (package)	95
\boxframe	180	color (package)	130, 613
breakurl (package)	577	\colorbox	73
bugs	154	\colorboxBlock	86, 9531
BVerbatim (environment)	222	colortbl (package)	614
bytefield (package)	578	CombineHigherDepths (boolean)	100, 307
C			
calc (package)	181	comment (package)	169
Calibre	142	Computer Modern	94
cancel (package)	578	\ConTeXt	10356
capt-of (package)	443	continue (package)	616
caption (package)	136, 186, 443, 579	converting	
caption2 (package)	582	class	152
\caption@begin	8010	document	93
\caption@end	8010	package	150
\captionlistentry	8014	counter:	
\captionof	8045	FileDepth	100, 307
\CaptionSeparator	7954	FootnoteDepth	100, 300
ccaption (package)	582	lofdepth	450
center (environment)	9439	lotdepth	450
\centering	9463	LWR@cellcolordepth	398
\centerline	9479	LWR@currentautosec	318
changebar (package)	583	LWR@externalfilecnt	462
changepage (package)	583	LWR@hdashedlines	363
chappg (package)	584	LWR@hlines	362
\chapter	5397	LWR@htmlfilenumber	275
chapterbib (package)	584	LWR@latestautopage	440
chemfig (package)	585	LWR@lateximagedepth	492
chemformula		LWR@lateximagenumber	492
troubleshooting	127	LWR@LIpage	492
chemformula (package)	586	LWR@midrulecounter	389
chemgreek (package)	591	LWR@minipagedepth	516
chemistry		LWR@nextautoid	440
Greek symbols	591	LWR@nextautopage	440
chemmacros (package)	592	LWR@nextequation	474
chemnum (package)	612	LWR@prevFileDepth	317
chnpage (package)	584	LWR@previousautopagelabel	309
cite (package)	613	LWR@spandepth	290
class		LWR@startingequation	494
modifying for lwarp	152	LWR@tablecolindex	365
class (key) [Gin]	676	LWR@tablecolspecindex	364
class:		LWR@tablecolspecwidth	364
internet	68	LWR@tabletotalcols	365
komascript	137	LWR@tabletotalcolsnext	365
memoir	138, 850	LWR@tabularDepth	363
		LWR@tabularpardepth	364
		LWR@thisautoid	438

LWR@thisautoidWP	438	\end@float	7917
SideTOCDepth	99, 448	endfloat (package)	623
tocdepth	99	endheads (package)	623
\cpagerefFor	9670	endnotes (package)	121, 624
crop (package)	616	\enlargethispage	10255
cross-references		\enskip	535, 10201
missing or incorrect	155	enumerate (environment)	6104
CSS		enumerate (package)	625
file selection	104	enumitem (package)	625
lwrap.css	105	environ (package)	182
per HTML page	104	environment:	
project-specific changes	105	fminipage	114
\CSSFilename	101, 104, 4808	lateximage	490
\CustomizeMathJax	5079, 5104	picture	515
cuted (package)	616	titlepage	107
cutwin (package)	617	titlingpage	107
		warpall	106
D		warpHTML	103, 106
danger icon	161	warpprint	102, 106
\date	108	environments:	
dblfloatfix (package)	618	abstract	5811
dblfnote (package)	618	align	9050
dcolumn (package)	618	align*	9076
debugging	154	alignat	9154
HTML debug comments	190	alignat*	9180
\DeclareGraphicsExtensions	2	BlockClass	4579
Deja Vu	94	BVerbatim	222
description		center	9439
HTML meta tag	108, 299	description	6123
description (environment)	6123	enumerate	6104
diagbox (package)	619	eqnarray	8885
display math		equation	8807
complicated objects	127	equation*	8815
\displaymathnormal	127, 461, 479, 8825	fcolorminipage	161, 9572
\displaymathother	127, 461, 479, 8836	flalign	9102
document		flalign*	9128
convert existing	93	flushleft	9455
\dotfill	10178	flushright	9447
\doublerulesepcolor	6600	fminipage	9923
\doublerulesepcolornexttoken	6600	gather	8998
draftwatermark (package)	620	gather*	9024
dynamic math expressions	272	itemize	6098
		lateximage	9309, 9434
E		list	6068
easy-todo (package)	621	longtable	2
ebook (package)	622	LWR@BlockClassWP	4616
ellipsis (package)	622	LWR@blocktextcurrentfont	10103
\emph	9985	LWR@displaymathnormal	8681
emptypage (package)	623	LWR@displaymathother	8686
\enddblfloat	7917	LWR@equationother	8700

floatpag (package)	652	fullpage (package)	668
floatrow (package)	137, 653	fullwidth (package)	668
fltrace (package)	658	\fup	10130
\flushbottom	4361	\fussy	4363
flushend (package)	659	fwlw (package)	668
flushleft (environment)	9455		
flushright (environment)	9447	G	
fminipage (environment)	114, 9923	gather (environment)	8998
fncychap (package)	659	gather* (environment)	9024
fnlineno (package)	660	GELLMU (program)	68
fnpos (package)	660	generator	
font		HTML meta tag	327
bitmapped	94	geometry (package)	179, 669
Computer Modern	94	getttitlestring (package)	180
Deja Vu	94	[Gin]:	
ligatures	95	class (key)	676
missing symbols	111	\Gin@setfile	101
packages	95	GladTeX (program)	68
selection	94	glossaries (package)	121, 669
size — lateximage	124, 459, 491	glossary	
size — math, SVG	124, 459, 491	language	90
size — xfrac	870	processing	90
fontenc (package)	95, 661	glossary (package)	671
fontspec (package)	95, 661	glyphtounicode.tex (file)	95
footmisc (package)	661	graphics	
footnote (package)	662	EPS files	129, 506
FootnoteDepth (counter)	100, 300	PDF files	129, 506
footnotehyper (package)	664	graphics (package)	129, 672
footnotes	299	graphicx (package)	129, 683
in math	118	Greek	
in sectioning commands	118	chemistry symbols	591
misnumbered	118	grffile (package)	130, 684
verbatim	118	grid (package)	684
footnpag (package)	664	gridset (package)	684
for			
HTML & PRINT	161	H	
HTML output	161	hang (package)	685
PRINT output	161	hanging (package)	686
\ForceHTMLPage	119, 5188	hash	
\ForceHTMLTOC	120, 5194	SVG image filename	464, 497
forest (package)	664	heading, word processor	147
FormatEpub (boolean)	142, 195	Hevea (program)	68
FormatWP (boolean)	144, 195	\hfill	10176
\framebox	9876	\HomeHTMLFilename	4247
framed (package)	665	HomeHTMLFilename (option)	97, 99, 168
framed objects	114	horizontal space	110
FrameMaker (program)	69	between minipages	535
Frequently Asked Questions	154	\href	7811
\frontmatter	5210	\hrulefill	10177
ftnright (package)	667	\hskip	111

`\hspace` ... [110](#), [535](#), [10209](#), [10239](#), [10245](#)
`htlatex` (program) [68](#)
HTML
 alt tags [127](#)
 appearing in svg images [85](#), [157](#)
 commands for a successful conver-
 sion [109](#)
 conversion settings [98](#)
 debug comments [190](#)
 EPUB [142](#), [195](#)
 word processor [144](#), [195](#)
 filename generation [103](#)
 headings [160](#)
 meta tag
 author [109](#), [299](#)
 description [108](#), [299](#)
 generator [327](#)
 Title [298](#)
 title [108](#), [298](#)
 viewport [327](#)
 selecting print/HTML definitions .. [190](#)
 tabular column conversion [378](#)
`\HTMLAuthor` [101](#), [109](#), [4834](#)
`HTMLDebugComments` (boolean) [190](#)
`\HTMLDescription` [101](#), [108](#), [4839](#)
`\HTMLentity` [4228](#)
`\HTMLFilename` [4246](#)
`HTMLFilename` (option) [97](#), [99](#), [169](#)
`\HTMLFirstPageTop` [101](#), [4796](#)
`htmlglossary` (option) [lwarpmk] [122](#), [670](#)
`\HTMLLanguage` [101](#), [5471](#)
`\HTMLleftmargini` (length)
 [139](#), [345](#), [346](#), [850](#)
`\HTMLPageBottom` [102](#), [334](#), [4802](#)
`\HTMLPageTop` [101](#), [4799](#)
`\HTMLTitle` [101](#), [108](#), [4829](#)
`\HTMLUnicode` [4236](#)
`\HTMLvleftskip` (length) [139](#), [345](#), [346](#), [850](#)
`hypcap` (package) [687](#)
`hypdestopt` (package) [688](#)
`\hyperindexref` [8269](#)
`hypernat` (package) [688](#)
`hyperref`
 incorrect links [156](#)
`hyperref` (package) [117](#), [431](#), [688](#)
`\hypertoc` [8171](#)
`\hypertocfloat` [8191](#)
`hyperxmp` (package) [694](#)
`hyphenat` (package) [694](#)

I

icon
 warning [161](#)
`idxlayout` (package) [695](#)
`\if@titlepage` [5623](#)
`ifoddpage` (package) [696](#)
`ifplatform` (package) [164](#)
images
 appearing as HTML [85](#), [157](#)
 EPS [129](#), [506](#)
 graphicx package [506](#)
 hashed filenames [464](#), [497](#)
 in strange places [156](#)
 PDF
 converting to SVG [91](#), [129](#)
 using [129](#), [506](#)
 processing [248](#)
`Improper \prevdepth.` [121](#), [154](#)
`\include` [671](#)
`\includegraphics` [230](#)
`indentfirst` (package) [696](#)
`InDesign` (program) [69](#)
index
 language [90](#)
 placement and toc options [122](#)
 processing [83](#), [84](#)
 UTF-8 [95](#)
`\InlineClass` [4604](#)
`inputenc` (package) [95](#), [697](#)
`inputenx` (package) [95](#), [697](#)
`internet` (class) [68](#)
item
 empty [111](#), [351](#)
`itemize` (environment) [6098](#)
`\itshape` [10116](#)

J

JavaScript
 MathJax [123](#), [458](#)

K

`\kern` [111](#)
key:
 [Gin]:
 class [676](#)
`keyfloat` (package) [137](#), [697](#)
`komascript` (class) [137](#)
`kvoptions` (package) [166](#)

L	
<code>\l@chapter</code>	8208
<code>\l@figure</code>	8215
<code>\l@paragraph</code>	8213
<code>\l@part</code>	8207
<code>\l@section</code>	8210
<code>\l@subparagraph</code>	8214
<code>\l@subsection</code>	8211
<code>\l@subsubsection</code>	8212
<code>\l@table</code>	8216
label	
in HTML	330
math environment	480
valid characters	117
Label(s) may have changed	154
language	
glossary	90
index	90
language HTML metadata	326
<code>\LaTeX</code>	10332
LaTeX2HTML (program)	68
<code>\LaTeXe</code>	10332
lateximage (environment) 490, 9309, 9434	
<code>\LateximageFontScale</code>	9218
<code>\LateximageFontSizeName</code>	9217
lateximages	
font size	124, 459, 491
processing	248
latexmk (option)	97, 99, 169
LaTeXML (program)	68
layout (package)	699
Leaders not followed by proper glue. .	154
<code>\leftline</code>	9478
length:	
<code>\HTMLleftmargini</code> .	139, 345, 346, 850
<code>\HTMLvleftskip</code> . .	139, 345, 346, 850
<code>\LWR@cmidrulewidth</code>	389
<code>\LWR@heavyrulewidth</code>	389
<code>\LWR@lightrulewidth</code>	389
<code>\LWR@minipageheight</code>	517
<code>\LWR@minipagewidth</code>	516
<code>\LWR@tempheight</code>	536
<code>\LWR@tempraise</code>	536
<code>\LWR@tempwidth</code>	536
<code>\LWR@thiscmidrulewidth</code>	389
<code>\VerbatimHTMLwidth</code>	346
<code>\vleftmargini</code>	139, 345, 850
<code>\vleftskip</code>	139, 345, 850
letltxmacro (package)	165
letterspace (package)	700
lettrine (package)	700
LibreOffice	
conversion recommendations	147
import into	144
section headings	147
LibreOffice (program)	69
ligatures	95, 179
line numbers	161
<code>\linebreak</code>	10249
lineno (package)	701
<code>\LinkHome</code>	4300
Linux (program)	105, 165
lips (package)	704
list	
empty item	111, 351
filename	112
list (environment)	6068
listings (package)	704
<code>\listof</code>	8132
<code>\listoffigures</code>	8106
<code>\listoftables</code>	8119
lmodern (package)	95
lofdepth (counter)	450
longtable (environment)	2
longtable (package)	134, 708
lotdepth (counter)	450
lscape (package)	710
ltablex (package)	710
ltxcaption (package)	711
ltxgrid (package)	711
ltxtable (package)	712
lua-check-hyphen (package)	712
luacolor (package)	712
LuaLaTeX	
detection	162
file & section names	314
<code>\LuaLaTeX</code>	10350
LuaLaTeX (program) [requirement] . . .	72
<code>\LuaTeX</code>	10350
luatodonotes (package)	140, 713
lwarp	
loading	96
options	96
lwarp (package)	96
lwarp-patch-komascript (package) .	879
lwarp-patch-memoir (package)	881
lwarp.css (file)	105, 205
lwarp.xdy	
customizing	109
lwarp.xdy (file)	109, 244

lwrap_baseline_marker.png (file) ..	463	LWR@colafterspec	365
lwrap_formal.css (file)	239	LWR@colatspec	365
lwrap_mathjax.txt (file)	245	LWR@colbangspec	365
lwrap_one_limage.cmd (file)	245	LWR@colbarspec	365
lwrap_sagebrush.css (file)	234	LWR@colbeforespec	365
lwrap_tutorial.txt (file)	79	\LWR@columnHTMLcolor	6593
lwrapmk		\LWR@columnspeclookahead	6346
customizing	152	\LWR@convertto	65
[lwrapmk]:		LWR@copiedsidetoc (boolean)	445
htmlglossary (option)	122, 670	\LWR@copyfile	706
printglossary (option)	122, 670	\LWR@createautosec	5219
lwrapmk (option)	97, 168, 248	LWR@currentautosec (counter)	318
lwrapmk (program)	152, 248	\LWR@currentcss	4807
lwrapmk.conf (file)	203	\LWR@currenttextcolor	10141
lwrapmk.lua (file)	152	\LWR@customizedMathJax	5078
\LWR@addbaselinemarker	8408	\LWR@customizeMathJax	5084
\LWR@addcdashline	6867	\LWR@descitem	6110
\LWR@addcmidruletrim	6833	LWR@displaymathnormal (environment)	
\LWR@addcmidrulewidth	6864	8681
\LWR@addformatwppalignment	6883	LWR@displaymathother (environment)	8686
\LWR@addleftmostbartag	6670	\LWR@docdashline	6797
\LWR@addmathjax	8749	\LWR@docmidrule	6781
\LWR@addmulticolvertrulecolor ..	7142	\LWR@doequation	8760
\LWR@addrulewidth	6838	LWR@doingapar (boolean)	290
\LWR@addtabularcellcolor	7011	LWR@doingcmidrule (boolean)	363
\LWR@addtabularhrulecolor	6911	LWR@doingstartpars (boolean)	290
\LWR@addtabulararrowcolor	6896	LWR@doingtbrule (boolean)	363
\LWR@addtabularrulecolors	6957	\LWR@domulticolumn	7180
\LWR@afterendverbatim	5894	\LWR@doubledollar	8599
\LWR@amsmathbody	9296	LWR@dynamicmath (boolean)	272
\LWR@amsmathbodynumbered	9302	\LWR@earlyloadnever	191
LWR@amsmultline (boolean)	480	LWR@emptyatbang (boolean)	363
\LWR@atbeginverbatim	5875	\LWR@endfloatalignment	7950
\LWR@backgroundcolor	36	\LWR@endhideamsmath	8878
\LWR@beginhideamsmath	8870	\LWR@endoffline	10184
\LWR@blackborderpadding	9894	\LWR@ensuredoingapar	4678
LWR@BlockClassWP (environment) ..	4616	LWR@equationother (environment) ..	8700
LWR@blocktextcurrentfont (environ-		\LWR@equationtag	9269
ment)	10103	LWR@exitingtabular (boolean)	414
\LWR@botnavigation	4306	\LWR@expandableformatted	768
\LWR@caption@begin	7989	\LWR@expandableformattedenv	793
\LWR@caption@end	8001	LWR@externalfilecnt (counter)	462
LWR@cdashlines	389	LWR@figcaption (environment)	7969
LWR@cellcolordepth (counter)	398	\LWR@filenamenoblanks	4990
\LWR@cellHTMLcolor	6595	\LWR@filestart	5478
\LWR@clearmidrules	6748	\LWR@findcurrenttextcolor	10144
\LWR@closeparagraph	4702	\LWR@findword	582
\LWR@closeprevious	4333	\LWR@floatalignment	7936
\LWR@closetabledatcell	6236	\LWR@floatalignmentname	7935
\LWR@cmidrulewidth (length)	389	\LWR@floatbegin	7868

\LWR@floatend	7902	LWR@intabularmetadata (boolean) ..	363
\LWR@floatstyle	2	LWR@isstartingequation (boolean) ..	495
\LWR@footnotetext	4856	\LWR@itemizeitem	6091
\LWR@forceminwidth	9883	LWR@latestautopage (counter)	440
\LWR@forcenewpage	4350	LWR@lateximagedepth (counter)	492
\LWR@formatted	753	\LWR@lateximagedepthref	7670
\LWR@formattedenv	783	LWR@lateximagenumber (counter)	492
\LWR@formatting	752	\LWR@lateximagenumberref	7673
LWR@freezethisautoid (boolean) ..	438	\LWR@lateximagesfile	668
\LWR@futurenonSPACELET	6182	\LWR@lightrulewidth (length)	389
\LWR@FVstyle	56	LWR@LIpage (counter)	492
\LWR@getexparray	4182	\LWR@listitem	6053
\LWR@getmynexttoken	6189	\LWR@loadafter	160
LWR@hdashedlines (counter)	363	\LWR@loadbefore	174
\LWR@heavyrulewidth (length)	389	\LWR@loadnever	183
\LWR@hidelatexequation	8735	\LWR@longtabledatacaptiontag ..	7240
LWR@hlines (counter)	362	\LWR@lookforpackagename	585
\LWR@htmlblockcomment	4529	\LWR@LwarpEnd	5599, 10386
\LWR@htmlblocktag	4531	\LWR@LwarpStart	5528, 10386
\LWR@HTMLcline	7512	\LWR@maketitlesetup	33, 5717
\LWR@htmlclosecomment	4501	LWR@mathmacro (boolean)	271
\LWR@htmlcomment	4522	\LWR@maybeewtablerow	6602
\LWR@htmldivclass	4556	\LWR@maybeprintpendingfootnotes	4930
\LWR@htmldivclassend	4559	LWR@midrulecounter (counter)	389
\LWR@htmlmlelement	4569	LWR@midrules	388
\LWR@htmlmlelementclass	4543	LWR@minipagedepth (counter)	516
\LWR@htmlmlelementclassend	4548	LWR@minipagefullwidth (boolean) ..	517
\LWR@htmlmlelementclassline	4562	\LWR@minipageheight (length)	517
\LWR@htmlmlelementend	4572	\LWR@minipagestartpars	10189
LWR@htmlfilenumber (counter)	275	\LWR@minipagestoppars	10192
\LWR@htmlfileref	7667	LWR@minipagethispar (boolean)	517
\LWR@HTMLhline	7495	\LWR@minipagewidth (length)	516
\LWR@htmlmathlabel	8855	\LWR@multicolother	7083
\LWR@htmlmulticolumn	7229	\LWR@multicolparttext	7077
\LWR@htmlopencomment	4501	\LWR@multicolskip	7089
\LWR@htmlrefsectionfilename	4290	\LWR@multirowborder	3
\LWR@HTMLsanitize	9221	\LWR@mynexttoken	6181
\LWR@HTMLsanitizeexpand	9238	\LWR@myshorttoc	8055
\LWR@htmlsectionfilename	4257	\LWR@nameref	7664
\LWR@htmlspan	4478	LWR@nestspan (environment)	4443
\LWR@htmlspanclass	4486	\LWR@newautoidanchor	7923
\LWR@htmltag	4496	\LWR@newautopagelabel	5070
\LWR@htmltagc	4432	\LWR@newhtmlfile	5107
\LWR@HTMLtextstyle	9980	\LWR@newlabel	7704
\LWR@imagesizebox	100	\LWR@newref	7755
\LWR@includegraphicsb	105	LWR@nextautoid (counter)	440
\LWR@indexitem	8241, 8245, 8249	LWR@nextautopage (counter)	440
LWR@indisplaymathimage (boolean) ..	462	LWR@nextequation (counter)	474
LWR@infloatrow (boolean)	676	\LWR@notmemoirloadafter	171
\LWR@instertatbangcols	6228	\LWR@nullfonts	10050

\LWR@nullifyNoAutoSpacing	7514	\LWR@rowHTMLcolor	6594
\LWR@nulllistfills	6060	\LWR@ruleHTMLcolor	6596
\LWR@opacity	99	\LWR@sanitize	4218
\LWR@openparagraph	4684	\LWR@sanitized	4217
\LWR@opttablecol (boolean)	379	\LWR@scalestyle	94
\LWR@origcolspec	6222	\LWR@section	5236
\LWR@originname	244	\LWR@sectionnumber	5216
\LWR@originnames	251	\LWR@select@html@hspace	10209
\LWR@origmathjax (boolean)	166	\LWR@select@html@nohspace	10239
\LWR@overline	10136	\LWR@select@print@hspace	10242
\LWR@parseaftercolumn	6416	\LWR@setexparray	4177
\LWR@parseatcolumn	6356	\LWR@setlatestname	7642
\LWR@parsebangcolumn	6382	\LWR@setOSWindows	73
\LWR@parsebarcolumn	6427	\LWR@sidetoc	8148
\LWR@parsebeforecolumn	6405	\LWR@singledollar	8623
\LWR@parsecoloncolumn	6452	\LWR@skipatbang (boolean)	363
\LWR@parseDcolumn	6498	\LWR@skippingmcolrowcell (boolean)	363
\LWR@parsedrequirepackagenames	581	\LWR@skippingmrowcell (boolean)	363
\LWR@parsenormalcolumn	6477	\LWR@spandepth (counter)	290
\LWR@parsepcolumn	6489	\LWR@splabel	7676
\LWR@parsesemicoloncolumn	6473	\LWR@starredlongtable (boolean)	406
\LWR@parsetablecols	6504	\LWR@startedrow (boolean)	362
\LWR@parsewcolumn	6493	\LWR@startingequation (counter)	494
\LWR@patchlists	6129	\LWR@startingequationtag	9268
\LWR@prevFileDepth (counter)	317	\LWR@startnewdepth	5226
\LWR@previousautopagelabel (counter)	309	\LWR@startpars	4751
\LWR@printatbang	6645	\LWR@startref	7717
\LWR@printbartag	6635	\LWR@stoppars	4769
\LWR@printcloselist	5960	\LWR@stripperiod	7641
\LWR@printlength	563	\LWR@strresult	6220
\LWR@printmccoldata	7092	\LWR@subaddcmidruletrim	6820
\LWR@printmccoltype	7035	\LWR@subaddtabularcellcolor	7005
\LWR@printopenlist	5961	\LWR@subcdashline	6784
\LWR@printpendingfootnotes	4920	\LWR@subcmidrule	6766
\LWR@printpendingmpfootnotes	4939	\LWR@subhtml elementclass	4536
\LWR@printthetitle	5669	\LWR@subhyperref	7787
\LWR@providelength	62	\LWR@subhyperrefclass	7800
\LWR@ProvidesPackageDrop	649	\LWR@subhyperrefreftext	7795
\LWR@ProvidesPackagePass	638	\LWR@subinlineimage	7861
\LWR@pushoneclose	5222	\LWR@sublabel	7681
\LWR@quickfile	665	\LWR@subnewref	7749
\LWR@ref@ignorestar	7761	\LWR@subsingledollar	8440
\LWR@remembertag	9273	\LWR@subtableofcontents	8068
\LWR@requesttoc	5592	\LWR@subtabularhtmlcolumns	7310
\LWR@requirepackagenames	580	\LWR@syncmathjax	8715
\LWR@restoreorigaccents	1019	\LWR@tablecolindex (counter)	365
\LWR@restoreorigformatting	8307	\LWR@tablecolspec	6219
\LWR@restoreoriglists	6157	\LWR@tablecolspecindex (counter)	364
\LWR@rotboxorigin	243	\LWR@tablecolspecwidth (counter)	364
\LWR@rotstyle	91	\LWR@tabledatacolumnntag	7399

<code>\LWR@tabledatasinglecolumnntag</code> ..	6684	<code>makeidx</code> (package)	122, 181, 831
<code>LWR@tableparcell</code> (boolean)	363	<code>\MakeIndex</code>	10359
<code>LWR@tabletotalcols</code> (counter)	365	<code>\makelabel</code>	351
<code>LWR@tabletotalcolsnext</code> (counter) ..	365	<code>\maketitle</code>	47, 107, 5729
<code>LWR@tabular</code> (environment)	7527	<code>margin</code>	
<code>LWR@tabularcelladded</code> (boolean) ...	362	numbers	161
<code>LWR@tabularDepth</code> (counter)	363	tags	161
<code>\LWR@tabularendoffline</code>	6328	<code>marginfit</code> (package)	715
<code>\LWR@tabularfinishrow</code>	6288	<code>marginfix</code> (package)	716
<code>\LWR@tabularhtmlcolumns</code>	7320	<code>marginnote</code> (package)	716
<code>\LWR@tabularleftedge</code>	6675	<code>\marginpar</code>	112, 305, 4951
<code>LWR@tabularmutemods</code> (boolean)	414	<code>\marginparBlock</code> ...	112, 305, 4962, 4979
<code>LWR@tabularpardepth</code> (counter)	364	<code>\markboth</code>	4358
<code>\LWR@tdaddstyle</code>	6808	<code>\markright</code>	4359
<code>\LWR@tdendstyles</code>	6814	<code>markup languages</code>	69
<code>\LWR@tdstartstyles</code>	6807	<code>math</code>	
<code>\LWR@tempcolor</code>	10142	<code>\displayingmath</code>	127
<code>\LWR@tempcolortwo</code>	10142	<code>\displayingmore</code>	127
<code>\LWR@tempheight</code> (length)	536	<code>tabbing</code>	131
<code>\LWR@tempraise</code> (length)	536	<code>mathjax option</code>	167
<code>\LWR@tempwidth</code> (length)	536	<code>mathsvg option</code>	167
<code>\LWR@textcurrentcolor</code>	19, 10147	<code>chemformula</code>	127
<code>\LWR@textcurrentfont</code>	10095	<code>mhchem</code>	728
<code>LWR@thisautoid</code> (counter)	438	<code>Tikz</code>	127
<code>LWR@thisautoidWP</code> (counter)	438	<code>appearing as HTML</code>	85, 157
<code>\LWR@thiscmidrulewidth</code> (length) ...	389	<code>display with complicated objects</code> ..	127
<code>\LWR@thiscolspec</code>	6683	<code>dynamic</code>	272
<code>\LWR@thisfilename</code>	4988	<code>font size — SVG</code>	124, 459, 491
<code>\LWR@thisnewfilename</code>	4989	<code>footnotes</code>	118
<code>\LWR@titlingmaketitle</code>	83, 5768	<code>images incorrect</code>	272
<code>\LWR@topnavigation</code>	4303	<code>labels — valid characters</code>	117
<code>\LWR@traceinfo</code>	734	<code>MathJax custom functions</code> ...	125, 460
<code>LWR@tracinglwarp</code> (boolean)	189	<code>MathJax summary</code>	125, 459
<code>LWR@trimlrules</code>	388	<code>SVG summary</code>	124, 459
<code>LWR@trimrrules</code>	389	<code>word processor conversion</code>	147
<code>LWR@validtablecol</code> (boolean)	377	<code>math</code> (environment)	8680
<code>LWR@verbtags</code> (boolean)	346	<code>\mathimagenam</code>	102, 8395
<code>\LWR@vspace</code>	10246	<code>MathJax</code>	
<code>\LWR@WPcell</code>	6879	<code>mathjax option</code>	167
<code>\LWR@xcolorrowHTMLcolor</code>	6592	<code>custom functions</code>	125, 460
<code>\LWRPrintStack</code>	4311	<code>errors</code>	127
<code>\LWRsetnextfloat</code>	7963	<code>subequations</code>	126, 460
<code>\LyX</code>	10367	<code>summary</code>	125, 459
		<code>tagged equations</code>	125, 460
		<code>MathJax</code> (program)	125, 459, 460
		<code>MathJax</code> (program) [requirement]	72
		<code>mathjax</code> (boolean)	166
		<code>mathjax</code> (option)	97, 99, 167
		<code>mathsvg</code> (option)	97, 99, 167
		<code>\mbox</code>	9851
M			
<code>Mac OS</code> (program)	105, 165		
<code>Madcap</code> (program)	69		
<code>magaz</code> (package)	715		
<code>\mainmatter</code>	5206		
<code>\makebox</code>	9854		

[tocbibind]:		
numindex	123, 831	
[tocloft]:		
titles	120	
BaseJobname	97, 167	
HomeHTMLFilename	97, 99, 168	
HTMLFilename	97, 99, 169	
latexmk	97, 99, 169	
lwarpmk	97, 168, 248	
mathjax	97, 99, 167	
mathsvg	97, 99, 167	
OSWindows	97, 105, 165, 168	
pdftotextEnc	97, 168	
warpHTML	96, 167	
warpprint	96, 166	
xindyCodepage	97, 122, 167, 670, 830	
xindyLanguage	90, 97, 122, 167, 670, 830	
xindyStyle	97, 109, 168	
options clash, memoir	138	
\OSPathSymbol	72	
OSWindows (option)	97, 105, 165, 168	
overpic (package)	131, 756	
P		
package		
modifying for lwarpmk	150	
required	178	
package:		
a4	547	
a4wide	547	
a5comb	547	
abstract	120, 547	
acro	550	
acronym	551	
addlines	554	
adjmulticol	553	
ae	554	
aecc	554	
afterpage	555	
algorithm2e	555	
algorithmicx	559	
algorithmx	136	
alltt	560	
amsmath	182	
amsthm	561	
anonchap	565	
anysize	565	
appendix	120, 566	
arabicfront	566	
array	567	
arydshln	567	
atbegshi	569	
authblk	120, 570	
axodraw2	571	
babel	139	
backref	571	
balance	571	
bibunits	572	
bigdelim	135, 572	
bigstrut	573	
blowup	574	
bookmark	574	
booktabs	575	
boxedminipage	576	
boxedminipage2e	577	
breakurl	577	
bytefield	578	
calc	181	
cancel	578	
capt-of	443	
caption	136, 186, 443, 579	
caption2	582	
ccaption	582	
changebar	583	
changepage	583	
chappg	584	
chapterbib	584	
chemfig	585	
chemformula	586	
chemgreek	591	
chemmacros	592	
chemnum	612	
chnngpage	584	
cite	613	
cleveref	117, 512	
clrdblpg	613	
cmap	95	
color	130, 613	
colortbl	614	
comment	169	
continue	616	
crop	616	
cuted	616	
cutwin	617	
dblfloatfix	618	
dblfnote	618	
dcolumn	618	
diagbox	619	
draftwatermark	620	
easy-todo	621	

ebook	622	fullpage	668
ellipsis	622	fullwidth	668
emptypage	623	fwlw	668
endfloat	623	geometry	179, 669
endheads	623	getttitlestring	180
endnotes	121, 624	glossaries	121, 669
enumerate	625	glossary	671
enumitem	625	graphics	129, 672
environ	182	graphicx	129, 683
epigraph	626	grffile	130, 684
epstopdf	129, 131, 506, 627	grid	684
epstopdf-base	628	gridset	684
errata	628	hang	685
eso-pic	630	hanging	686
etoolbox	164	hypcap	687
everyhook	180	hypdestopt	688
everypage	631	hypernat	688
everyshi	631	hyperref	117, 431, 688
expl3	180	hyperxmp	694
extramarks	631	hyphenat	694
fancybox	115, 632	idxlayout	695
fancyhdr	638	ifoddpage	696
fancyheadings	638	ifplatform	164
fancyref	639	indentfirst	696
fancyvrb	640	inputenc	95, 697
figcaps	646	inputenx	95, 697
figsize	647	keyfloat	137, 697
filecontents	180	kvoptions	166
fix2col	647	layout	699
fixme	140, 647	letltxmacro	165
fixmetodonotes	649	letterspace	700
flafter	649	lettrine	700
float	136, 650	lineno	701
floatflt	651	lips	704
floatpag	652	listings	704
floatrow	137, 653	lmodern	95
fltrace	658	longtable	134, 708
flushend	659	lscape	710
fncychap	659	ltablex	710
fnlineno	660	ltxcaption	711
fnpos	660	ltxgrid	711
fontenc	95, 661	ltxtable	712
fontspec	95, 661	lua-check-hyphen	712
footmisc	661	luacolor	712
footnote	662	luatodonotes	140, 713
footnotehyper	664	lwrap	96
footnpag	664	lwrap-patch-komascript	879
forest	664	lwrap-patch-memoir	881
framed	665	magaz	715
ftnright	667	makeidx	122, 181, 831

marginfit	715	printlen	182
marginfix	716	quotchap	764
marginnote	716	quoting	765
mcaption	717	ragged2e	766
mdframed	116, 717	realscripts	766
memhfixc	727	refcount	181
metalogo	728	register	767
mhchem	728	relsize	112, 768
microtype	179, 731	resizegather	769
midfloat	731	romanbar	770
midpage	732	romanbarpagenumber	770
morefloats	732	rotating	770
moreverb	732	rotfloat	771
morewrites	734	savetrees	772
mparhack	734	scalefmt	772
multicol	734	schemata	772
multirow	736	scrextend	773
multitoc	739	scrhack	776
nameref	739	scrlayer	776
natbib	739	scrlayer-notecolumn	778
needspace	740	scrlayer-scrpage	778
newclude	139, 741	section	779
newfloat	181	sectionbreak	780
newtxmath	128	sectsty	781
newunicodechar	95, 741	setspace	782
nextpage	741	shadow	783
nicefrac	128, 742	showidx	783
nonfloat	742	showkeys	783
nonumonpart	743	sidecap	784
nopageno	743	sidenotes	785
nowidow	743	siunitx	128, 505, 786
ntheorem	128, 744	soul	792
overpic	131, 756	soulpos	794
pagegrid	757	soulutf8	794
pagenote	121, 757	stabular	795
pagesel	758	stfloats	795
paralist	758	subcaption	136
parnotes	759	subfig	136, 796
parskip	760	subfigure	801
pbox	760	supertabular	135, 802
pdfscape	761	syntonly	803
pdfrender	761	tlenc	803
pdfsync	761	tabls	804
pfnote	762	tabularx	804
phfqit	762	tabulary	805
placeins	763	textarea	805
polyglossia	140	textcomp	95, 111, 805
prelim2e	763	textfit	809
prettyref	763	textpos	810
preview	763	theorem	811

threeparttable	815	zref	182
tikz	130, 816	zwpagelayout	878
titles	817	\packagediagramname	102, 8396
titleref	820	page	
titlesec	821	counter	85, 157
titletoc	823	inaccessible	100
titling	120, 825	\pagebreak	10251
tocbasic	830	\pagecolor	71
tocbibind	123, 830, 831	pagegrid (package)	757
toccenter	832	pagenote (package)	121, 757
tocloft	120, 122, 565, 830, 833	\pagenumbering	4364
tocstyle	839	\pageref	7765
todo	840	\pagerefPageFor	7764
todonotes	140, 841	pagesel (package)	758
transparent	843	\pagestyle	4356
trimclip	843	Pandoc (program)	69
trivfloat	136, 844	\par	
turnthepage	845	hooks	295
typearea	845	\paragraph	5453
ulem	846	paralist (package)	758
underscore	848	\parbox	9842
units	128	parnotes (package)	759
upref	848	\parsemulticolumnalignment	7127
url	117, 849	parskip (package)	760
varioref	117	\part	5384
varwidth	183	pbox (package)	760
verbatim	181	PDF images	
verse	138, 849, 850	converting to SVG	91, 129
vertbars	851	using	129, 506
vmargin	852	pdfcrop (program) [requirement]	72
vowel	852	pdfLaTeX (program) [requirement]	72
vwcol	853	pdfscape (package)	761
wallpaper	855	pdfrender (package)	761
wasysym	856	pdfseparate (program) [requirement]	72, 77
watermark	856		72, 77
wrapfig	856	pdfsync (package)	761
xcolor	130, 507, 858	pdftocairo (program)	129, 506
xellipsis	869	pdftocairo (program) [requirement]	72, 77
xfrac	870	pdftotext (program) [requirement]	72, 77
xifthen	181	pdftotextEnc (option)	97, 168
xltable	873	Perl	78
xltxtra	873	perl (program) [requirement]	78
xmpinl	874	pfnote (package)	762
xparse	141, 180	\phantomsection	10312
xpatch	164	phfqit (package)	762
xpiano	874	picture (environment)	515, 9734
xstring	181	placeins (package)	763
xtab	135, 875	Plastex (program)	68
xurl	877	\PN@parnotes@auto	4683
xy	877	polyglossia (package)	140

<code>\popclose</code>	4150	<code>TeX2page</code>	68
<code>Poppler</code>	72, 77	<code>TeX4ht</code>	68
<code>prelim2e</code> (package)	763	<code>TeXMaths</code>	147
<code>prettyref</code> (package)	763	<code>TtH</code>	68
<code>preview</code> (package)	763	<code>Unix</code>	105, 165
<code>print</code>		<code>Windows</code>	105, 165
selecting print/HTML definitions ..	190	<code>Word</code>	69
<code>\printauthor</code>	334, 5677, 5696	<code>xindy</code>	109
<code>\printdate</code>	334, 5688, 5698	<code>programs</code>	
<code>printglossary</code> (option) [lwarpmk] ..	122, 670	utility	71
<code>\printindex</code>	8224	<code>project.css</code> (file)	105
<code>printlen</code> (package)	182	<code>project.lwarpmkconf</code> (file)	204
<code>\printthanks</code>	333, 336	<code>project_html.tex</code> (file)	202
<code>\printtitle</code>	334, 5661, 5695	<code>published</code>	342
<code>problems</code>	154	<code>\pushclose</code>	4123
<code>program:</code>			
[requirement]:		Q	
LuaLaTeX	72	<code>\qqquad</code>	535, 10200
MathJax	72	<code>\quad</code>	535, 10195
pdfcrop	72	<code>quotchap</code> (package)	764
pdfLaTeX	72	<code>quote</code> (environment)	5837
pdfseparate	72, 77	<code>quoting</code> (package)	765
pdftocairo	72, 77		
pdftotext	72, 77	R	
perl	78	<code>ragged2e</code> (package)	766
XeLaTeX	72	<code>\raggedbottom</code>	4360
<code>Adobe</code>	69	<code>\raggedleft</code>	9468
<code>AsciiDoc</code>	69	<code>\raggedright</code>	9473
<code>AsciiDoctor</code>	69	<code>\raisebox</code>	9971
<code>Asciidoctor-LaTeX</code>	69	<code>realscripts</code> (package)	766
<code>Flare</code>	69	<code>\ref</code>	7755
<code>FrameMaker</code>	69	<code>refcount</code> (package)	181
<code>GELLMU</code>	68	<code>\reflectbox</code>	297
<code>GladTeX</code>	68	<code>register</code> (package)	767
<code>Hevea</code>	68	<code>relsize</code> (package)	112, 768
<code>htlatex</code>	68	[requirement]:	
<code>InDesign</code>	69	LuaLaTeX (program)	72
<code>LaTeX2HTML</code>	68	MathJax (program)	72
<code>LaTeXML</code>	68	pdfcrop (program)	72
<code>LibreOffice</code>	69	pdfLaTeX (program)	72
<code>Linux</code>	105, 165	pdfseparate (program)	72, 77
<code>lwarpmk</code>	152, 248	pdftocairo (program)	72, 77
<code>Mac OS</code>	105, 165	pdftotext (program)	72, 77
<code>Madcap</code>	69	perl (program)	78
<code>MathJax</code>	125, 459, 460	XeLaTeX (program)	72
<code>MS-Windows</code>	105, 165	<code>\RequirePackage</code>	613
<code>OpenOffice</code>	69	<code>\resizebox</code>	306
<code>Pandoc</code>	69	<code>resizegather</code> (package)	769
<code>pdftocairo</code>	129, 506	<code>\ResumeTabular</code>	7386
<code>Plastex</code>	68	<code>\reversemarginpar</code>	4975

<code>\rightline</code>	9480	<code>\sidetocname</code>	8145
<code>\rmfamily</code>	10112	<code>\simplechapterdelim</code>	5233
<code>romanbar</code> (package)	770	<code>siunitx</code>	
<code>romanbarpagenumber</code> (package)	770	with TeXMaths	147
<code>\rotatebox</code>	263	<code>siunitx</code> (package)	128, 505, 786
<code>rotating</code> (package)	770	<code>\sloppy</code>	4362
<code>rotfloat</code> (package)	771	<code>soul</code> (package)	792
<code>\rowcolor</code>	6597	<code>soulpos</code> (package)	794
<code>\rownum</code>	6589	<code>soulutf8</code> (package)	794
<code>\rule</code>	10258	<code>\sp</code>	10119
S			
<code>sample_project.css</code> (file)	105, 244	<code>space</code>	
<code>savetrees</code> (package)	772	horizontal	110
<code>\sb</code>	10120	between minipages	535
<code>\scalebox</code>	282	<code>split</code>	
<code>scalefnt</code> (package)	772	miss-numbered	128, 744
<code>schemata</code> (package)	772	<code>stabular</code> (package)	795
<code>scrextend</code> (package)	773	<code>stack depths</code>	160
<code>scrhack</code> (package)	776	<code>\StartDefiningMath</code>	4195
<code>scrlayer</code> (package)	776	<code>\StartDefiningTabulars</code>	4185
<code>scrlayer-notecolumn</code> (package)	778	<code>\StartDynamicMath</code>	4207
<code>scrlayer-scrpage</code> (package)	778	<code>stfloats</code> (package)	795
<code>\scshape</code>	10117	<code>\StopDefiningMath</code>	4199
<code>section</code>		<code>\StopDefiningTabulars</code>	4189
depths	160	<code>\StopDynamicMath</code>	4211
heading, word processor	147	<code>subcaption</code> (package)	136
missing	100	<code>subequations</code>	
<code>\section</code>	5423	MathJax	126, 460
<code>section</code> (package)	779	<code>subfig</code> (package)	136, 796
<code>sectionbreak</code> (package)	780	<code>subfigure</code> (package)	801
<code>sectsty</code> (package)	781	<code>\subparagraph</code>	5461
<code>\SetHTMLFileNumber</code>	4248	<code>\subsection</code>	5436
<code>setspace</code> (package)	782	<code>\subsubsection</code>	5444
<code>settings</code>		<code>subtitle</code>	342
css project-specific	105	<code>supertabular</code> (package)	135, 802
CSS selection	104	<code>SVG</code>	
HTML conversion	98	mathsvg option	167
lwrap package options	96	converting from PDF	91, 129
selecting output	105	dynamic math	272
title page	107	image processing	248
<code>\sffamily</code>	10113	images appearing as HTML ...	85, 157
<code>\sfrac</code>	870	math incorrect images	272
<code>shadow</code> (package)	783	math summary	124, 459
<code>shell escape</code>	94	<code>syntonly</code> (package)	803
<code>showidx</code> (package)	783	T	
<code>showkeys</code> (package)	783	<code>tlenc</code> (package)	803
<code>sidecap</code> (package)	784	<code>tabbing</code>	
<code>sidenotes</code> (package)	785	and math	131
<code>SideTOCDepth</code> (counter)	99, 448	<code>tabbing</code> (environment)	5921

table		\thempfootnote	4917
text alignment	136	theorem (package)	811
\tableofcontents	102, 8087	\thetitle	334
tabls (package)	804	\thispagestyle	4357
tabular		threeparttable (package)	815
baseline	378	tikz	
HTML columnn conversion	378	catcodes	816
macros inside	133, 360	dollar redefined	816
misplaced \noalign	134, 360	in display math	127
misplaced alignment tab character &	132, 358	tikz (package)	130, 816
multicolumn with multirow	738	Title	
row corruption	133, 360	HTML meta tag	298
\TabularMacro	7384	title	
tabularx (package)	804	HTML meta tag	108, 298
tabulary (package)	805	\title	107, 4818
tagged equations		titlepage	
Mathjax	125, 460	subtitle and published	342
\TeX	10320	titlepage (environment)	107, 5645
TeX2page (program)	68	titleps (package)	817
TeX4ht (program)	68	titleref (package)	820
TeXMaths (program)	147	titles (option) [tocloft]	120
textarea (package)	805	titlesec (package)	821
\textbf	9994	titletoc (package)	823
\textcolor	48	titling (package)	120, 825
textcomp		titlingpage (environment)	14, 107
missing symbols	111	tocbasic (package)	830
textcomp (package)	95, 111, 805	[tocbibind]:	
textfit (package)	809	numindex (option)	123, 831
\textgreater	4238	tocbibind (package)	123, 830, 831
\textit	10019	tocdepth (counter)	99
\textless	4238	tocenter (package)	832
\textmd	9988	[tocloft]:	
\textnormal	10036	titles (option)	120
textpos (package)	810	tocloft (package)	120, 122, 565, 830, 833
\textrm	9997	tocstyle (package)	839
\textsc	10022	todo (package)	840
\textsf	10004	todonotes (package)	140, 841
\textsl	10029	\tracinglwrap	158, 733
\textsubscript	10123	transparent (package)	843
\textsuperscript	10121	trimclip (package)	843
\texttt	10009	trivfloat	
\textup	10012	with newfloat, algorithmic	845
\tfl@chapter@fix	844	trivfloat (package)	136, 844
\thanks	108	troubleshooting	154
\thanksmarkseries	91	HTML debug comments	190
\theauthor	334	\ttfamily	10114
thebibliography (environment)	8288	TtH (program)	68
\thedate	334	turnthepage (package)	845
theindex (environment)	8231	tutorial.tex (file)	79
		typearea (package)	845

U		warning	
ulem (package)	846	Improper \prevdepth.	121, 154
\underline	10131	Label(s) may have changed.	154
underscore		Leaders not followed by proper glue.	154
filename	99, 112	Missing \$ inserted.	154
underscore (package)	848	Options clash, memoir	138
Unicode		warning icon	161
enhanced coverage	94	warpall (environment)	106, 149
file & section names	314	warpHTML (environment)	95, 103, 106, 150
input characters	163	warpHTML (option)	96, 167
selection	94	\warpHTMLonly	103, 106, 139
units (package)	128	warpingHTML (boolean)	166
Unix (program)	105, 165	warpingprint (boolean)	166
\up	10129	warpprint (environment)	102, 106, 150
upref (package)	848	warpprint (option)	96, 166
\upshape	10115	\warpprintonly	103, 106, 138
\url	7844	wasysym (package)	856
url (package)	117, 849	watermark (package)	856
usingOSWindows (boolean)	165	Windows (program)	105, 165
UTF-8		Word (program)	69
enhanced coverage	94	word processor	
file & section names	314	conversion recommendations	147
index	95	HTML conversion settings	144, 195
selection	94	section headings	147
utility		WPMarkFloats (boolean)	146, 195
programs	71	WPMarkLOFT (boolean)	147, 196
V		WPMarkMath (boolean)	147, 196
varioref (package)	117	WPMarkMinipages (boolean)	146, 195
varwidth (package)	183	WPMarkTOC (boolean)	146, 196
verbatim		WPTitleHeading (boolean)	147, 196
footnotes	118	wrapfig (package)	856
verbatim (environment)	5911	X	
verbatim (package)	181	xcolor (package)	130, 507, 858
\VerbatimHTMLWidth (length)	346	XeLaTeX	
\verbatiminput	5903	detection	162
verse (environment)	2, 5852	file & section names	314
verse (package)	138, 849, 850	\XeLaTeX	10352
vertbars (package)	851	XeLaTeX (program) [requirement]	72
viewport		xellipsis (package)	869
HTML meta tag	327	\XeTeX	10352
\vleftmargini (length)	139, 345, 850	xfrac (package)	870
\vleftskip (length)	139, 345, 850	\xfracHTMLfontsize	3
vmargin (package)	852	xifthen (package)	181
vowel (package)	852	xindy	
vwcol (package)	853	customizing	109
W		xindy (program)	109
wallpaper (package)	855	xindyCodepage (option)	
			97, 122, 167, 670, 830

xindyLanguage (option)	xpatch (package)	164
..... 90, 97, 122, 167, 670, 830	xpiano (package)	874
xindyStyle (option)	xstring (package)	181
..... 97, 109, 168	xtab (package)	135, 875
xltabular (package)	xurl (package)	877
..... 873	xy (package)	877
xltextra (package)		
..... 873		
xmpincl (package)		
..... 874		
xparse		
	Z	
warnings	zref (package)	182
..... 141	zwpagelayout (package)	878
xparse (package)		
..... 141, 180		

For the most recent changes and the start of the Index, see page 929.