



The **lwarp** package

ℒ_AT_EX to HTML

v0.40 — 2017/09/25

© 2016–2017 Brian Dunn
bd@BDTechConcepts.com

Abstract

The **lwarp** package allows ℒ_AT_EX to directly produce HTML output, using external utility programs only for the final conversion of text and images. Math may be represented by svg files or MathJax.

Documents may be produced by pdfℒ_AT_EX, Luaℒ_AT_EX, or Xeℒ_AT_EX. A `texlua` script removes the need for system utilities such as `make` and `gawk`, and also supports `xindy` and `latexmk`. Configuration is automatic at the first manual compile.

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

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

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

[To update existing projects, see section 1, Updates.](#)

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

License:

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

Contents

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

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

7.6	Graphics	90
7.6.1	grffile package	91
7.6.2	color package	91
7.6.3	xcolor package	91
7.7	Tabular	92
7.7.1	longtable package	94
7.7.2	supertabular and xtab packages	94
7.7.3	bigdelim package	95
7.8	Floats	95
7.8.1	float, trivfloat, and/or algorithmicx together	95
7.8.2	caption and subcaption packages	95
7.8.3	subfig package	95
7.8.4	floatrow package	96
7.9	Miscellaneous	96
7.9.1	verse and memoir	96
7.9.2	newclude package	97
7.9.3	babel package	97
8	EPUB conversion	98
9	Word-processor conversion	100
10	Modifying lwarp	102
10.1	Creating an lwarp version of a package	102
10.2	Adding a package to the lwarp.dtx file	103
10.3	Testing lwarp	103
10.4	Modifying lwarpmk	103
11	Troubleshooting	105
11.1	Using the lwarp.sty package	105
11.1.1	Debug tracing output	107
11.2	Compiling the lwarp.dtx file	107
12	Implementation	108
13	Stack depths	108
14	Source Code	110
15	Detecting the T_EX Engine — pdf_latex, lua_latex, xel_latex	111
16	Unicode Input Characters	111
17	Early package requirements	112
18	Operating-System portability	113
18.1	Common portability code	113

18.2	Unix, Linux, and Mac OS	113
18.3	MS-Windows	113
19	Package options	114
19.1	Conditional compilation	117
20	Misplaced packages	118
21	Required packages	122
22	Loading packages	127
23	File handles	130
24	Include a file	130
25	Copying a file	131
26	Debugging messages	132
27	HTML-conversion output modifications	133
28	Remembering original formatting macros	134
29	Accents	135
30	Configuration Files	137
30.1	project_html.tex	137
30.2	lwarpmk.conf	137
30.3	project.lwarpmkconf	138
30.4	lwarp.css	139
30.5	lwarp_sagebrush.css	165
30.6	lwarp_formal.css	170
30.7	sample_project.css	175
30.8	lwarp.xdy	175
30.9	lwarp_mathjax.txt	176
30.10	lwarpmk option	178
31	Stacks	189
31.1	Assigning depths	189
31.2	Closing actions	190
31.3	Closing depths	190
31.4	Pushing and popping the stack	191
32	Data arrays	192
33	HTML entities	193

34	HTML filename generation	194
35	Homepage link	196
36	\PrintStack diagnostic tool	197
37	Closing stack levels	198
38	PDF pages and styles	198
39	HTML tags, spans, divs, elements	199
39.1	Mapping \LaTeX Sections to HTML Sections	200
39.2	Babel-French	200
39.3	HTML tags	201
39.4	Block tags and comments	203
39.5	Div class and element class	205
39.6	Single-line elements	206
39.7	HTML5 semantic elements	206
39.8	High-level block and inline classes	207
39.9	Closing HTML tags	208
40	Paragraph handling	208
41	Paragraph start/stop handling	212
42	Page headers and footers	214
43	CSS	216
44	HTML meta description and author	216
45	Footnotes	217
45.1	Regular page footnotes	217
45.2	Minipage footnotes	217
45.3	Titlepage thanks	218
45.4	Regular page footnote implementation	218
45.5	Minipage footnote implementation	220
45.6	Printing pending footnotes	220
46	Marginpars	221
47	Splitting HTML files	221
48	Sectioning	227
48.1	User-level starred section commands	227
48.2	Book class commands	228
48.3	Sectioning support macros	228
48.4	\section and friends	235

49	Starting a new file	237
50	Starting HTML output	239
51	Ending HTML output	242
52	Title page	244
52.1	Setting the title, etc.	244
52.2	Changes for <code>\affiliation</code>	246
52.3	Printing the thanks	247
52.4	Printing the title, etc. in HTML	247
52.5	Printing the title, etc. in print form	248
52.6	<code>\maketitle</code> for HTML output	249
52.7	<code>\published</code> and <code>\subtitle</code>	252
53	Abstract	253
54	Quote and verse	254
54.1	Citations and attributions	254
54.2	Quotes, quotations	254
54.3	Verse	255
55	Verbatim	256
56	Theorems	258
57	Lists	259
57.1	Itemize	260
57.2	Enumerate	260
57.3	Description	261
58	Tabular	262
58.1	Limitations	262
58.2	Token lookahead	265
58.3	Booleans	266
58.4	Handling <code>&</code> , <code>@</code> , and <code>!</code>	266
58.4.1	Localizing <code>&</code> catcodes	268
58.4.2	Handling <code>&</code>	268
58.5	Handling <code>\</code>	269
58.6	Variables	270
58.7	Parsing <code>@</code> , <code>></code> , <code><</code> , <code>!</code> columns	271
58.8	Parsing <code>'l'</code> , <code>'c'</code> , or <code>'r'</code> columns	274
58.9	Parsing <code>'p'</code> , <code>'m'</code> , or <code>'b'</code> columns	274
58.10	Parsing <code>'D'</code> columns	275
58.11	Parsing the column specifications	275
58.12	Starting a new row	279
58.13	Printing at or bang tags	280

58.14	Data opening tag	280
58.15	Midrules	282
58.16	Multicolumns	283
58.16.1	Parsing multicolumns	283
58.16.2	Multicolumn factored code	286
58.16.3	Multicolumn	287
58.16.4	Longtable captions	287
58.16.5	Counting HTML tabular columns	290
58.17	Multicolumnrow	291
58.18	Utility macros inside a table	293
58.19	Special-case tabular markers	293
58.20	Checking for a new table cell	294
58.21	\mrowcell	296
58.22	New \tabular definition	296
58.23	Array	300
59	Cross-references	300
59.1	Setup	300
59.2	Zref setup	302
59.3	Labels	303
59.4	References	305
59.5	Hyper-references	307
60	Floats	310
60.1	Float captions	311
60.1.1	Caption inside a float environment	312
60.1.2	Caption and LOF linking and tracking	313
61	Table of Contents, LOF, LOT	316
61.1	Reading and printing the TOC	316
61.2	High-level TOC commands	318
61.3	Side TOC	319
61.4	Low-level TOC line formatting	320
62	Index and glossary	323
63	Math	325
63.1	Inline and display math	327
63.2	MathJax support	329
63.3	Equation environment	333
63.4	AMS Math environments	335
63.4.1	Support macros	335
63.4.2	Environment patches	336
64	Lateximages	343
65	center, flushleft, flushright	349

66	Pre-loaded packages	350
67	Siunitx	351
68	Graphics	351
68.1	xcolor print-mode additions	352
69	Cleveref	356
70	Picture	358
71	Boxes and Minipages	358
71.1	Counters and lengths	359
71.2	Footnote handling	359
71.3	Minipage handling	360
71.4	Parbox, makebox, framebox, fbox, raisebox	363
72	Direct formatting	368
73	Skips, spaces, font sizes	373
74	\phantomsection	380
75	\LaTeX and other logos	380
75.1	HTML logos	381
75.2	Print logos	382
76	\AtBeginDocument, \AtEndDocument	383
77	Trademarks	384
2	lwarp-a4.sty	385
78	a4	385
3	lwarp-a4wide.sty	385
79	a4wide	385
4	lwarp-a5comb.sty	385
80	a5comb	385

5	lwarp-abstract.sty	385
81	abstract	385
6	lwarp-adjmulticol.sty	387
82	adjmulticol	387
7	lwarp-addlines.sty	388
83	addlines	388
8	lwarp-afterpage.sty	389
84	afterpage	389
9	lwarp-algorithmicx.sty	389
85	algorithmicx	389
10	lwarp-alltt.sty	390
86	alltt	390
11	lwarp-amsthm.sty	390
87	amsthm	390
12	lwarp-anonchap.sty	393
88	anonchap	393
13	lwarp-anysize.sty	393
89	anysize	393
14	lwarp-appendix.sty	393
90	appendix	393

15	lwarp-arabicfront.sty	394
91	arabicfront	394
16	lwarp-authblk.sty	394
92	authblk	394
17	lwarp-balance.sty	395
93	balance	395
18	lwarp-bigdelim.sty	395
94	bigdelim	395
19	lwarp-bigstrut.sty	397
95	bigstrut	397
20	lwarp-bookmark.sty	397
96	bookmark	397
21	lwarp-booktabs.sty	398
97	booktabs	398
22	lwarp-boxedminipage.sty	398
98	boxedminipage	398
23	lwarp-boxedminipage2e.sty	398
99	boxedminipage2e	398
24	lwarp-caption2.sty	399
100	caption2	399

25	lwarp-ccaption.sty	399
101	ccaption	399
26	lwarp-changepage.sty	399
102	changepage	399
27	lwarp-chngpage.sty	400
103	chngpage	400
28	lwarp-chappg.sty	400
104	chappg	400
29	lwarp-color.sty	400
105	color	400
30	lwarp-crop.sty	400
106	crop	400
31	lwarp-cuted.sty	401
107	cuted	401
32	lwarp-cutwin.sty	401
108	cutwin	401
33	lwarp-dblfnote.sty	402
109	dblfnote	402
34	lwarp-dcolumn.sty	402
110	dcolumn	402

35	lwarp-draftwatermark.sty	403
111	draftwatermark	403
36	lwarp-ebook.sty	403
112	ebook	403
37	lwarp-ellipsis.sty	404
113	ellipsis	404
38	lwarp-emptypage.sty	404
114	emptypage	404
39	lwarp-endnotes.sty	404
115	endnotes	404
40	lwarp-enumerate.sty	405
116	enumerate	405
41	lwarp-enumitem.sty	405
117	enumitem	405
42	lwarp-epigraph.sty	406
118	epigraph	406
43	lwarp-eso-pic.sty	407
119	eso-pic	407
44	lwarp-everypage.sty	407
120	everypage	407

45	lwarp-everyshi.sty	408
121	everyshi	408
46	lwarp-extramarks.sty	408
122	extramarks	408
47	lwarp-fancybox.sty	408
123	fancybox	408
48	lwarp-fancyhdr.sty	412
124	fancyhdr	412
49	lwarp-fancyvrb.sty	413
125	fancyvrb	413
50	lwarp-figcaps.sty	418
126	figcaps	418
51	lwarp-fix2col.sty	419
127	fix2col	419
52	lwarp-float.sty	419
128	float and \newfloat	419
53	lwarp-floatflt.sty	420
129	floatflt	420
54	lwarp-floatpag.sty	421
130	floatpag	421

55	lwarp-floatrow.sty	421
131	floatrow	421
56	lwarp-flushend.sty	427
132	flushend	427
57	lwarp-fncychap.sty	427
133	fncychap	427
58	lwarp-fnpos.sty	428
134	fnpos	428
59	lwarp-fontenc.sty	428
135	fontenc	428
60	lwarp-fontspec.sty	429
136	fontspec	429
61	lwarp-footmisc.sty	429
137	footmisc	429
62	lwarp-footnote.sty	430
138	footnote	430
63	lwarp-footnotehyper.sty	431
139	footnotehyper	431
64	lwarp-framed.sty	431
140	framed	431

65	lwarp-ftnright.sty	434
141	ftnright	434
66	lwarp-fullpage.sty	434
142	fullpage	434
67	lwarp-fullwidth.sty	434
143	fullwidth	434
68	lwarp-geometry.sty	435
144	geometry	435
69	lwarp-glossaries.sty	435
145	glossaries	435
70	lwarp-graphics.sty	437
146	graphics	437
	146.1 Graphics extensions	437
	146.2 Length conversions and graphics options	438
	146.3 \includegraphics	441
71	lwarp-graphicx.sty	447
147	graphicx	447
72	lwarp-grffile.sty	447
148	grffile	447
73	lwarp-hyperref.sty	447
149	hyperref	447

74	lwarp-hyperxmp.sty	452
150	hyperxmp	452
75	lwarp-idxlayout.sty	452
151	idxlayout	452
76	lwarp-indentfirst.sty	453
152	indentfirst	453
77	lwarp-inputenc.sty	453
153	inputenc	453
78	lwarp-keyfloat.sty	453
154	keyfloat	453
79	lwarp-layout.sty	454
155	layout	454
80	lwarp-letterspace.sty	455
156	letterspace	455
81	lwarp-lettrine.sty	455
157	lettrine	455
82	lwarp-lips.sty	456
158	lips	456

83	lwarp-listings.sty	457
159	listings	457
84	lwarp-longtable.sty	460
160	longtable	460
85	lwarp-lscape.sty	462
161	lscape	462
86	lwarp-ltcaption.sty	462
162	ltcaption	462
87	lwarp-marginfit.sty	463
163	marginfit	463
88	lwarp-marginfix.sty	463
164	marginfix	463
89	lwarp-marginnote.sty	463
165	marginnote	463
90	lwarp-mcaption.sty	464
166	mcaption	464
91	lwarp-mdframed.sty	464
167	mdframed	464

92	lwarp-metalogo.sty	473
168	metalogo	473
93	lwarp-microtype.sty	473
169	microtype	473
94	lwarp-midfloat.sty	474
170	midfloat	474
95	lwarp-moreverb.sty	474
171	moreverb	474
96	lwarp-mparhack.sty	476
172	mparhack	476
97	lwarp-multicol.sty	476
173	multicol	476
98	lwarp-multirow.sty	477
174	multirow	477
174.1	Multirow	478
174.2	Combined multicolumn and multirow	479
99	lwarp-nameref.sty	480
175	nameref	480
100	lwarp-needspace.sty	480
176	needspace	480

101	lwarp-newclude.sty	480
177	newclude	480
102	lwarp-newunicodechar.sty	481
178	newunicodechar	481
103	lwarp-nextpage.sty	481
179	nextpage	481
104	lwarp-nonumonpart.sty	481
180	nonumonpart	481
105	lwarp-nopageno.sty	482
181	nopageno	482
106	lwarp-nowidow.sty	482
182	nowidow	482
107	lwarp-ntheorem.sty	482
183	ntheorem	482
108	lwarp-pagenote.sty	493
184	pagenote	493
109	lwarp-paralist.sty	493
185	paralist	493
110	lwarp-parskip.sty	494
186	parskip	494

111	lwarp-pdfscape.sty	494
187	pdfscape	494
112	lwarp-pdfsync.sty	494
188	pdfsync	494
113	lwarp-pfnote.sty	495
189	pfnote	495
114	lwarp-placeins.sty	495
190	placeins	495
115	lwarp-prelim2e.sty	495
191	prelim2e	495
116	lwarp-quotchap.sty	496
192	quotchap	496
117	lwarp-ragged2e.sty	497
193	ragged2e	497
118	lwarp-realscripts.sty	498
194	realscripts	498
119	lwarp-reysize.sty	499
195	reysize	499

120	lwarp-romanbar.sty	500
196	romanbar	500
121	lwarp-romanbarpagenumber.sty	500
197	romanbarpagenumber	500
122	lwarp-rotating.sty	501
198	rotating	501
123	lwarp-rotfloat.sty	501
199	rotfloat	501
124	lwarp-savetrees.sty	502
200	savetrees	502
125	lwarp-scalefnt.sty	502
201	scalefnt	502
126	lwarp-sectsty.sty	502
202	sectsty	502
127	lwarp-setspace.sty	503
203	setspace	503
128	lwarp-shadow.sty	504
204	shadow	504
129	lwarp-showidx.sty	505
205	showidx	505

130	lwarp-showkeys.sty	505
206	showkeys	505
131	lwarp-sidecap.sty	505
207	sidecap	505
132	lwarp-sidenotes.sty	506
208	sidenotes	506
133	lwarp-siunitx.sty	508
209	siunitx	508
134	lwarp-soul.sty	508
210	soul	508
135	lwarp-stabular.sty	510
211	stabular	510
136	lwarp-subfig.sty	510
212	subfig	510
137	lwarp-supertabular.sty	516
213	supertabular	516
138	lwarp-syntonly.sty	517
214	syntonly	517
139	lwarp-tables.sty	518
215	tables	518

140	lwarp-tabularx.sty	518
216	tabularx	518
141	lwarp-tabulary.sty	519
217	tabulary	519
142	lwarp-textarea.sty	519
218	textarea	519
143	lwarp-textcomp.sty	520
219	textcomp	520
144	lwarp-textpos.sty	523
220	textpos	523
145	lwarp-theorem.sty	523
221	theorem	523
146	lwarp-threeparttable.sty	527
222	threeparttable	527
147	lwarp-tikz.sty	528
223	tikz	528
148	lwarp-titleps.sty	529
224	titleps	529
149	lwarp-titleref.sty	532
225	titleref	532

150	lwarp-titlesec.sty	532
226	titlesec	532
151	lwarp-titletoc.sty	534
227	titletoc	534
152	lwarp-titling.sty	535
228	titling	535
153	lwarp-tocbibind.sty	540
229	tocbibind	540
154	lwarp-tocloft.sty	542
230	tocloft	542
155	lwarp-transparent.sty	547
231	transparent	547
156	lwarp-trivfloat.sty	548
232	trivfloat	548
	232.1 Combining \newfloat, \trivfloat, and algorithmicx	549
157	lwarp-typearea.sty	549
233	typearea	549
158	lwarp-ulem.sty	549
234	ulem	549

159	lwarp-upref.sty	551
235	upref	551
160	lwarp-verse.sty	551
236	verse	551
161	lwarp-wallpaper.sty	553
237	wallpaper	553
162	lwarp-wrapfig.sty	554
238	wrapfig	554
163	lwarp-xcolor.sty	555
239	xcolor	555
239.1	Xcolor definitions: location and timing	556
239.2	Code	558
164	lwarp-xfrac.sty	563
240	xfrac	563
165	lwarp-xtxtra.sty	565
241	xtxtra	565
166	lwarp-xmpincl.sty	566
242	xmpincl	566
167	lwarp-xtab.sty	566
243	xtab	566

168	lwarp-zwpagelayout.sty	568
244	zwpagelayout	568
	Change History and Index	570

List of Figures

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

List of Tables

1	\LaTeX -HTML generation — lwarp package — Supported functions	37
2	Required software programs	45
3	Files created along with the print version	54
4	Package options	71
5	Section depths and HTML headings	109
6	Tabular baseline	275
7	Tabular HTML column conversions	276
8	Cross-referencing data structures	301
9	Float data structures	310

File 1 **lwarp.sty**

1 Updates

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

For a detailed list of changes, see the Change History on page 570.

v0.40:

graphics, graphicx

`\includegraphics path`

△ image file extensions

bigdelim

symbols

fixes

margins

columns

footnotes

tabular

sectioning

- graphics and graphicx have been moved from the lwarp core, and are only loaded if requested with `\usepackage`.
- Improved graphics `\graphicspath` support. Multiple image directories may now be used. Refer to **.pdf files without a file extension** to allow the HTML version to use a `.svg`, `.png`, `.jpg`, or `.gif` version instead. See section 7.6.
- grffile is now directly supported instead of emulated.
- Fix for bigdelim, and improved documentation. See section 94.
- Improved \TeX and textcomp symbols.
- Fix for \TeX logos and `\InlineClass`, etc. inside a `lateximage`.
- Fix for xltextra with \TeX .
- Fixes for tocbibind with `\simplechapter`, etc.
- Fixes for `\multicolumnrow` and `\nullfont`s with older versions of `multitrow` and `xparse`.
- Added `\underline`.
- Added `adjmulticol`.
- Added `cuted`, `midfloat`.
- Added `pfnote`, `fnpos`, `dblfnote`.
- Added `stabular`, `tabs`.
- Added `sectsty`, `anonchap`, `quotchap`.

v0.39:

title pages

⚠ `\published and`
`\subtitle`

⚠ load order

tabular

multi column/row cell

⚠ macros inside tabular

⚠ tabular defined inside
 another environment

tabular

margins

page layout

- Improved the titlepage HTML code, `\thanks` notes, and `\maketitle`. titling is no longer required, but is still supported. The `\published` and `\subtitle` fields are no longer provided, but `\AddSubtitlePublished` replicates them using titling. See section 52.7. `authblk` is added, and should be loaded before titling. See section 52.
- `\multirow` now supports the new optional `vpos` argument.
- Added `\multicolumnrow` for combined `\multicolumn` and `\multirow`. See section 174.2.
- Tabular special cases:
 - Added `\TabularMacro` to mark custom macros inside tabular data cells, avoiding row corruption. See section 7.7.
 - Added `\ResumeTabular` for use when a tabular environment is defined inside another environment. See section 7.7.
- Added `supertabular`, `xtab`, `bigstrut`, `bigdelim`.
- Added `fullwidth`.
- Added `addlines`, `anysize`, `a4`, `a4wide`, `a5comb`, `textarea`, `zwpagelayout`, `typearea`, `ebook`.

v0.38:

forced single-pass compile

starred sections

updated tutorial

packages

font size

page numbering

front & back matter

- Added `lwarpmk print1` and `lwarpmk html1` actions to force a compile of the project a single time. Useful when multiple passes are not needed, or changes were not detected.
- Added `\ForceHTMLPage` and `\ForceHTMLTOC` to force a starred sectional unit onto its own HTML page and with its own TOC entry. See section 7.4.1.
- Modified the tutorial to use the new `\ForceHTMLPage` and `\ForceHTMLTOC` macros.
- Added `appendix`, `tocbibind`, `fncychap`, `fix2col`.
- Added `relsize`, `scalegnt`.
- Added `realscripts`, `metalogo`, `xltxtra`.
- Added `grffile`, `romanbar`.
- Added `arabicfront`, `chappg`, `nonumonpart`, `nopageno`, `romanbarpagenumber`.
- Docs: Improved description of the use of front/back matter. See section 7.4.
- Fix: `color` requests `xcolor`.
- Fix: `\part` for `article` class.

v0.37:

- `\include` for HTML
 - `latexmk`
 - accents and symbols
 - babel-french
- `\include` now maintains independent `.aux` files for HTML versions.
 - `comment`, used by `lwarp`, now maintains independent cut files for print and HTML versions, helping `latexmk` to better know whether to recompile.
 - Improved support for \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 7)
 - Improved source formatting.
 - `\fbox` and related now use `\fboxsep` and `\fboxrule`.
 - `\makebox` and `\framebox` now use width and position.
 - `\fcolorbox` and related now work inside a `lateximage`.
 - babel-french: Improvements for French variants, load order, footnotes, ellipses.
 - Improved footnote numbering. `lateximage` footnotes now appear as regular footnotes to match the numbering of the print version. Also fixed a regression with MathJax.
 - Improved `siunitx` units.
 - Fix for filenames while using MathJax.
 - Fix for `\rule` when `xcolor` is not loaded.
 - Added `transparent`, `upref`.

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

v0.34:

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


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

 instead of:


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

 This change makes it more consistent with \TeX standards, and avoids problems with space between arguments.
 - ⚠ Optional arguments
 - Likewise, `\InlineClass`'s optional argument now comes before the mandatory arguments:


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

spans with minipages

framing minipages

lateximage, SVG math,
tabular

eqnarray

verbatim packages

framing packages

list packages

babel-french

- Improved compatibility between spans, minipages, lists, frames, and math. Handles minipages and lists inside an HTML span, such as an `\fbox` containing a minipage, although with minimal HTML formatting. See section 7.2.2. `\fboxBlock` is added to frame minipages, tables, and lists with full HTML formatting but no longer inline, and behaves as `\fbox` for print output. The `fminipage` environment is added for framed minipages, as an environment with full HTML formatting, and draws a framed minipage in print output. See section 7.2.4. `\fbox` and minipages now often work in SVG math and `lateximages`. MathJax supports `\fbox`, but not `\fboxBlock` nor `fminipage`.

- Improved compatibility between `lateximage` and `minipage`, `\parbox`, `\makebox`, `\fbox`, `\framebox`, `\raisebox`, `\scalebox`, `\reflectbox`, `tabular`, `booktabs`.

- Improved font control for `lateximage`s and `SVG math`.

- Added the `eqnarray` environments.

- `fancyvrb` is no longer required (preloaded), but is still supported.

- Added `verbatim` and `moreverb`.

- Added `fancybox`, `boxedminipage2e` and `shadow`.

- `enumitem` is no longer required, but is still supported.

- Added `enumerate` and `paralist`.

- `titleps` is no longer required, but is still supported.

- Added `crop`.

- Added `rotfloat`, `marginfit`, and several minor packages; see the change log.

- Adds fixed-width HTML spaces around punctuation when using `babel-french`. Lua \TeX does not yet use the extra punctuation spacing.

v0.33:

- `Tabular @` and `!` columns now have their own HTML columns.
- `&` catcode changes are localized, perhaps causing errors about the tab alignment character `&`, so any definitions of macros or environments which themselves contain `tabular` and `&` must be enclosed within `\StartDefiningTabulars` and `\EndDefiningTabulars`. See section 58.4.1. This change is not required for the routine use of tables, but only when a table is defined inside another macro or environment, and while also using the `&` character inside the definition. This may include the use inside conditional expressions.
- Several math environments were incorrectly placed inline. Also, for `amsmath` with `SVG math`, the `fleqn` option has been removed, resulting in improved spacing for aligned equations.
- Bug fixes; see the changelog.

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

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

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

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

v0.30:

`lwarp-newproject`

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

To modify existing documents, remove from the document source:

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

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

HTML setup changes.

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

Old Macro	New Package Option
<code>\HomeHTMLFileName</code>	<code>HomeHTMLfilename</code>
<code>\HTMLFileName</code>	<code>HTMLfilename</code>
<code>\useLatexmk</code>	<code>latexmk</code>
<code>\warpOSwindows</code>	<code>OSWindows</code>

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

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

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

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

- The file `lwarp_html.xdy` has been renamed `lwarp.xdy`. To update each document's project:
 1. Make the changes shown above.
 2. Recompile the document in print mode. This updates the project's configuration files, and also generates the new file `lwarp.xdy`.
 3. The old file `lwarp_html.xdy` may be deleted.
- The new lwarp package option `xdyFilename` may be used to tell `lwarpmk` to use a custom `.xdy` file instead of `lwarp.xdy`. See section 6.10.
- Improvements in index processing:
 - `xindy`'s language is now used for index processing as well as glossary.
 - Print mode without `latexmk` now uses `xindy` instead of `makeindex`.
 - `texindy/xindy` usage depends on `pdflatex` vs `xelatex`, `lualatex`.
 - For `pdflatex` and `texindy`, the `-C utf8` option is used. This is supported in modern distributions, but a customized `lwarpmk.lua` may need to be created for use with older distributions.

v0.29:

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

v0.28:

- `\HTMLAuthor {<name>}` assigns HTML meta author if non-empty. Defaults to `\theauthor`.
- Boolean `HTMLDebugComments` controls whether HTML comments are added for closing `<div>`s, opening and closing sections, etc.
- Boolean `FormatEPUB` changes HTML output for easy EPUB conversion via an external program. Removes per-file headers, footers, and nav. Adds footnotes per chapter/section.
- Boolean `FormatWordProcessor` changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments.

- Boolean `HTMLMarkFloats` adds text marks around floats only if `FormatWordProcessor`. These make it easier to identify float boundaries, which are to be manually converted to word-processor frames.
- Updated for the new MathJax CDN repository.
- Adds tabulary.
- Supports the options syntax for graphics.
- Improved index references, now pointing exactly to their target.
- Adds glossaries. `lwarpmk` is modified to add `printglossary` and `htmlglossary` actions.

2 Introduction

The lwarp project aims to allow a rich \LaTeX document to be converted to a reasonable HTML interpretation. No attempt has been made to force \LaTeX to provide for every HTML-related possibility, and HTML cannot exactly render every possible \LaTeX concept. Where compromise is necessary, it is desirable to allow the print output to remain typographically rich, and compromise only in the HTML conversion.

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

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

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

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

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

The lwarp package allows \LaTeX to directly generate HTML tags from a \LaTeX source document, with only minor intervention on the user’s part. A `texlua` program called `lwarpmk` is used to process either the print or HTML version of the document. A few external utility programs are used to finish the conversion from a \LaTeX -generated PDF file which happens to have HTML tags, to a number of HTML plain-text files and accompanying images.

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

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

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

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

Enough functionality is provided to convert a typical article containing technical content. Not every package has been tested, but many of the most useful ones are known to work, either as-is or through emulation with replacement code. (See table 1 on page 37.)

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

2.1 Supported packages and features

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

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

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

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

lwarp Supported Functions — continued

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

lwarp Supported Functions — continued

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

lwarp Supported Functions — continued

Category	Status
multicol:	Emulated, with <code>css</code> . Converted to up to three columns with an optional heading, per browser support. Single-column if unsupported.
fullwidth:	Emulated.
Direct formatting:	<code>\emph</code> , <code>\textsuperscript</code> , <code>\textbf</code> , etc are supported. <code>\bfseries</code> , etc. are not yet supported. <code>lettrine</code> , <code>ulem</code> , <code>soul</code> , <code>reysize</code> , <code>scalegnt</code> , and <code>realscripts</code> are supported.
Ordinals:	<code>nth</code> , <code>fntcount</code> , and <code>engord</code> are supported.
Text ligatures:	Ligatures for symbols are supported. Ligatures for <code>f</code> , <code>q</code> , <code>t</code> are intentionally turned off because many simpler browsers do not display them correctly. Modern full-featured browsers re-create these ligatures on-the-fly.
Horizontal space:	HTML output for <code>thin-unbreakable</code> , <code>unbreakable</code> , <code>\enskip</code> , <code>\quad</code> , <code>\qqquad</code> , <code>\hspace</code> .
Rules:	<code>\rule</code> with <code>width</code> , <code>height</code> , <code>raise</code> , <code>text color</code> .
HTML reserved characters:	<code>\&</code> , <code>\textless</code> , and <code>\textgreater</code> are converted to HTML entities.
Fonts:	Used as-is. Appear in math expressions or embedded image environments.
Symbols:	Native \TeX diacriticals, <code>textcomp</code> .
Working as-is:	Various utility, calculation, file, and text-only packages, such as <code>calc</code> , <code>fileerr</code> , <code>somedefs</code> , <code>trace</code> , <code>xspace</code> .
Where:	
Supported:	The existing \TeX package is used.
Emulated:	The \TeX package is not used, but some/all of its functions are emulated. Null functions, lengths, and counters are provided for source compatibility.

3 Alternatives

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

3.1 Internet class

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

3.2 TeX4ht

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

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

3.3 Translators

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

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

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

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

Prog `LaTeXML` **\LaTeX XML**: <http://dlmf.nist.gov/LaTeXML/>

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

Prog `LaTeX2HTML` **\LaTeX 2HTML**: <http://www.latex2html.org/>
and <http://ctan.org/pkg/latex2html>.

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

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

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

3.4 AsciiDoc

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

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

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

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

Asciidoctor-LaTeX:

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

3.5 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.)

3.6 Word processors

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

3.7 Commercial systems

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™.

3.8 Comparisons

AsciiDoc, Pandoc, and various other markup languages typically have a syntax which tries to be natural and human-readable, but the use of advanced features tends to require many combinations of special characters, resulting in a complicated mess of syntax. By contrast, \LaTeX spells things out in readable words but takes longer to type, although integrated editors exist which can provide faster entry and a graphic user interface. For those functions which are covered by the typical markup language it is arguable that \LaTeX is comparably easy to learn, while \TeX provides many more advanced features where needed, along with a large number of pre-existing packages which provide solutions to numerous common tasks.

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

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

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

4 Installation

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

4.1 Installing the lwarp package

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

Pre-installed: Try entering into a command line:

```
Enter ⇒ kpsewhich lwarp.sty
```

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

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

```
Enter ⇒ tlmgr install lwarp
```

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

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

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

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

\TeX Live:

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

Mi \TeX :

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

This should be something like:

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

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

```
Enter ⇒ mktexlsr
```

— or —

```
Enter ⇒ texhash
```

Table 2: Required software programs

Provided by your \LaTeX distribution:

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

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

The lwarp package: This package.

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

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

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

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

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

Poppler PDF utilities:

pdftotext: Used to convert PDF to text.

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

pdftocairo: Used to convert images to SVG.

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

From Poppler: poppler.freedesktop.org.

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

```
Enter ⇒ brew install poppler
```

For Windows, see:

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

Perl:

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

perl.org, strawberryperl.com

Automatically downloaded from the internet as required:

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

From: mathjax.org

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

CTAN .dtx and .ins files: Another form of T_EX package is the .dtx and .ins source files. These files are used to create the documentation and .sty files.

1. See <http://ctan.org/pkg/lwarp> for the lwarp package.
 2. Download the zip archive lwarp.zip into your own lwarp directory.
 3. Unpack lwarp.zip.
 4. Locate the contents lwarp.dtx and lwarp.ins
 5. Create the documentation:
Enter ⇒ `pdflatex lwarp.dtx`
(several times)
 6. Create the .sty files:
Enter ⇒ `pdflatex lwarp.ins`
 7. Copy the .sty files somewhere such as the T_EX Live local tree found in the previous CTAN TDS section, under the subdirectory:
`<texlocal>/tex/latex/local/lwarp`
 8. Copy the documentation lwarp.pdf to a source directory in the local tree, such as:
`<texlocal>/doc/local/lwarp`
 9. Renew the cache:
Enter ⇒ `mktxlsr`
— or —
Enter ⇒ `texhash`
- Or, for Windows MiKTeX, start the program called MiKTeX Settings (Admin) and click on the button called Refresh FNDB.
10. See section 4.2.1 to generate your local copy of lwarpmk.
 11. Once the local version of lwarpmk.lua is installed, it may be made available system-wide as per section 4.2.

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

Just testing!

4.2 Installing the lwarpmk utility

(Note: If `lwarpmk` is not already installed, it is easiest to use a local copy instead of installing it system-wide. See section 4.2.1.)

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

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

Otherwise, continue with the following:

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

```
Enter ⇒ kpsewhich lwarpmk.lua
```

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

4. Create `lwarpmk`:

Unix: Create a symbolic link and make it executable:

- (a) Locate the TeX 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 TeX Live: Create a new `lwarpmk.exe` file:

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

Windows MiKTeX: Create a new `lwarpmk.bat` file:

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

4.2.1 Using a local copy of `lwarpmk`

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

1. When compiling the tutorial in section 5, use the `lwarpmk` option for the `lwarp` package:

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

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

Unix:

- (a) Make `lwarpmk.lua` executable:
Enter ⇒ `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.

4.3 Installing additional utilities

To test for the existence of the additional utilities:

Enter the following in a command line. If each programs' version is displayed, then that utility is already installed. See table 2 on page 45.

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

To install xindy, latexmk, and pdfcrop:

The TeX utilities xindy, latexmk, and pdfcrop may be provided by your operating system's package manager, and are also provided by the CTAN archive:

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

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

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

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

To install the Poppler utilities to a MacOS machine:

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

To install the Poppler utilities to a Windows machine:

1. See table 2 on page 45.
2. Download and extract the Poppler utilities pdftotext, pdfseparate, and pdftocairo to a directory, such as Poppler.
3. In the Start window, type "Path" to search for results related to Path. Or, open the control panel and search for "Path".
4. Choose "Edit the system environment variables" in the control panel.
5. Choose the "Environment Variables" button.

6. Choose the "Path" variable, then the "Edit" button.
7. Choose the "New" button to make an additional entry.
8. Enter the bin directory of the Poppler utilities, such as:
C:\Users\
Be sure to include \bin.
9. Click "Ok" when done.

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

1. Download and install a version of Perl, such as Strawberry Perl, to a directory without a space in its name, such as C:\Strawberry.
2. Edit the Path as seen above for the Poppler utilities.
3. Enter the bin directory of the Perl utility, such as:
C:\Strawberry\perl\bin
Be sure to include \bin.
4. Click "Ok" when done.

Any utilities installed by hand must be added to the PATH.

5 Tutorial

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

5.1 Starting a new project

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

File `tutorial.tex`

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

Copy from the documentation PDF:

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

Copy from the lwarp documentation directory:

Another copy may be found by entering into a command line:

```
Enter ⇒ texdoc -l lwarp_tutorial.txt
```

This should be in the `doc/latex/lwarp/` directory along with this PDF documentation. Copy `lwarp_tutorial.txt` directly into your `tutorial` directory, renamed as `tutorial.tex`.

File `lwarp_tutorial.txt`

⚠ Note: `.txt` suffix!

⚠ Bad formatting!

When using Windows, use an editor other than Notepad, since Notepad does not accept the end-of-line from a Unix text file.

3. Compile the project:

```
Enter ⇒ pdflatex tutorial.tex
```

(several times)

(`xelatex` or `lualatex` may be used as well.)

4. View the resulting `tutorial.pdf` with a PDF viewer.

A number of new files are created when `tutorial.tex` is compiled, as shown in table 3. These files are created by the lwarp package.

(Two of the new files are configuration files for the helper program `lwarpmk`. Whenever a print version of the document is created, the configuration files for `lwarpmk` are updated to record the operating system, \TeX program (`pdflatex`, `xelatex`, or `lualatex`), the filenames of the source code and HTML output, and whether the additional helper program `latexmk` will be used to compile the document.)

Figure 1: tutorial.tex listing

Note: There are two pages!

```

% Save this as tutorial.tex for the lwarp package tutorial.

\documentclass{book}

\usepackage{iftex}

% --- LOAD FONT SELECTION AND ENCODING BEFORE LOADING LWARP ---

\ifPDFTeX
\usepackage{lmodern}           % pdflatex
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\else
\usepackage{fontspec}         % XeLaTeX or LuaLaTeX
\fi

% --- LWARP IS LOADED NEXT ---
\usepackage[
% HomeHTMLFilename=index,      % Filename of the homepage.
% HTMLFilename={node-},        % Filename prefix of other pages.
% IndexLanguage=english,      % Language for xindy index, glossary.
% latexmk,                     % Use latexmk to compile.
% OSWindows,                   % Force Windows. (Usually automatic.)
% mathjax,                     % Use MathJax to display math.
]{lwarp}
% \boolfalse{FileSectionNames} % If false, numbers the files.

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

% --- LATEX AND HTML CUSTOMIZATION ---
\title{The Lwarp Tutorial}
\author{Some Author}
\setcounter{tocdepth}{2} % Include subsections in the \TOC.
\setcounter{secnumdepth}{2} % Number down to subsections.
\setcounter{FileDepth}{1} % Split \HTML\ files at sections
\booltrue{CombineHigherDepths} % Combine parts/chapters/sections
\setcounter{SideTOCDepth}{1} % Include subsections in the side\TOC
\HTMLAuthor{Some Author} % Sets the HTML meta author tag.
\HTMLLanguage{en-US} % Sets the HTML meta language.
\HTMLDescription{A description.} % Sets the HTML meta description.
\HTMLFirstPageTop{Name and \fbox{HOMEPAGE LOGO}}

```

```

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

\begin{document}

\maketitle % Or titlepage/titlingpage environment.

% An article abstract would go here.

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

\chapter{First chapter}

\section{A section}

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

\subsection{A subsection}

See \cref{fig:withtext}.

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

\section{Some math}

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

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

\end{document}

```

Table 3: Files created along with the print version

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

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

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

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

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

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

5.2 Compiling the print version with `lwarpmk`

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

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

1. Re-compile the print version:

```
Enter ⇒ lwarpmk print
```

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

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

```
Enter ⇒ lwarpmk print
```

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

4. Force a recompile to occur.

```
Enter ⇒ lwarpmk again
```

```
Enter ⇒ lwarpmk print
```

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

5. Process the index.⁴⁵

```
Enter ⇒ lwarpmk printindex
```

6. Recompile again to include the index.

```
Enter ⇒ lwarpmk print
```

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

```
Enter ⇒ lwarpmk print1
```

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

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

⁴A `lwarpmk printglossary` command is also available to process a glossary produced with the glossaries package. See section 7.4.8.

⁵Also see section 7.4.9 for index options.

5.3 Compiling the HTML version with lwarpmk

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

1. Compile the HTML version:

```
Enter ⇒ lwarpmk html
```

- (a) `lwarpmk` uses \LaTeX to process `tutorial_html.tex` to create `tutorial_html.pdf`.
- (b) `pdftotext` is then used to convert to the file `tutorial_html.html`. This file is a plain-text file containing HTML tags and content for the entire document.
- (c) `lwarpmk` manually splits `tutorial_html.html` into individual HTML files according to the HTML settings. For this tutorial, the result is `tutorial.html` (the home page), along with `First-chapter.html`⁶, `Some-math.html`, and the document's index in `_Index.html`.⁷

2. View the homepage in a web browser.

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

math

Note that math is still displayed as its plain-text \LaTeX source until the images of the math expressions have been generated. Math may be displayed as SVG images or by a MathJax script, as seen in sections 5.4 and 5.5.

3. Force a recompile:

```
Enter ⇒ lwarpmk again
```

```
Enter ⇒ lwarpmk html
```

```
Enter ⇒ lwarpmk print
```

4. Process the HTML index and recompile:⁸⁹

```
Enter ⇒ lwarpmk htmlindex
```

```
Enter ⇒ lwarpmk html
```

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

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

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

```
Enter ⇒ lwarpmk html1
```

⁶`First-chapter.html` also contains the first section, even though the second section is its own HTML page. This behavior is controlled by the boolean `CombineHigherDepths`.

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

⁸A `lwarpmk htmlglossary` command is also available to process a glossary produced with the `glossaries` package. See section 7.4.8.

⁹Also see section 7.4.9 for index options.

5.4 Generating the SVG images

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

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

1. Create the svg images:

```
Enter ⇒ lwarpmk limages
```

```
Enter ⇒ lwarpmk html
```

2. Move to the tutorial's math page and reload.
3. The math images are displayed using the same font and formatting as the printed version.
4. Copy/paste a math expression into a text editor to see the \LaTeX source.

 **Adding/removing** When a math expression, `picture`, or `Tikz` environment is added or removed, the svg images must be re-created with `lwarpmk limages` to maintain the proper image file sequence numbers.

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

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

 **Lots of files!**

5.5 Using MathJax for math

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

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

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

2. Recompile

```
Enter ⇒ lwarpmk html
```

3. Reload the math page.

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

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

5.6 Changing the CSS style

`\CSSFilename` `\CSSFilename` may be used to choose which `.css` file is used to display each section of the web page. Use `\CSSFilename` before `\begin{document}` to assign the style of the home page. If different parts of the website should have different styles, call `\CSSFilename` again before each section heading which creates a new file.

The styles provided by `lwarp` include:

`lwarp.css`: A default style if `\CSSFilename` is not used. This style is comparable to a plain \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 webpage.

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



Rename it!

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

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

5.7 Customizing the HTML output

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

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

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

Enter \Rightarrow `lwarpmk clearall`

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

Enter \Rightarrow `lwarpmk print`

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

Enter \Rightarrow `lwarpmk html`

Options for the `lwarp` package:

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

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

`HomeHTMLFilename=index`

[filename underscores](#)

causing the homepage to be the file `index.html`. Underscores are allowed in `HomeHTMLFilename` and `HTMLFilename` options, but may need to be escaped elsewhere, such as when appearing in a list:

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

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

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

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

Opt `mathjax` **mathjax:** Selects MathJax for math output.

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

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

Ctr SideTOCDepth

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

sideroc

Each subpage of the website has its own small table of contents on the side (the “sideroc”). Its depth is set by SideTOCDepth. This sideroc is only shown if the web page is wide enough. When using a narrow web browser window, “responsive web design” is used to show the sideroc at the top of the page and a link back to “Home” at the bottom.

It is recommended to set:

```
SideTOCDepth = FileDepth
```

or

```
SideTOCDepth = FileDepth+1
```

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



Ctr FileDepth

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

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



Bool CombineHigherDepths

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

The file splits are controlled by the counter FileDepth and the boolean CombineHigherDepths. Setting FileDepth to 0 splits the file at chapters, 1 at sections, etc. CombineHigherDepths controls whether to combine pages at levels higher than the chosen FileDepth, such as in this tutorial where the page which opens the chapter also contains the first section. Be careful to set tocdepth and SideTOCDepth to allow access to each page of the website. Set tocdepth and SideTOCDepth to be greater than or equal to FileDepth.

Inaccessible pages!

Lost in an old page!

When making changes to the file structure, it is possible to end up with the web browser pointing to an old file which is no longer in use. When this occurs, changes to the web site will not appear in the browser, even if reloading the page, because that page is no longer in use. It is best to return to the home page, clean the files (`lwarpmk cleanall`), change

FileDepth and/or CombineHigherDepths, then finally recompile and renavigate to the desired page using the new file structure.

Bool FileSectionNames

FileSectionNames: If true, web page filenames are derived from a sanitized version of the section names. If false, web pages are numbered. Either way, the HTMLFilename option is used as a prefix.

HTML filenames

Example HTML filenames:

Numbered HTML nodes:

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

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

Named HTML sections, no prefix:

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

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

Named HTML sections, with prefix:

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

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

\abstractname

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

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

\CSSFilename

\CSSFilename: `{\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.

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

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

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

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

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

Placed in the document wherever necessary:

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

Env <code>warpHTML</code>	warpHTML: An environment which is only used while generating HTML output. This is useful for website logos and other items which have no purpose in printed output.
<code>\warpprintonly</code>	\warpprintonly: <code>{\langle contents \rangle}</code> A macro version of the <code>warpprint</code> environment.
<code>\warpHTMLonly</code>	\warpHTMLonly: <code>{\langle contents \rangle}</code> A macro version of the <code>warpHTML</code> environment.

5.8 Using latexmk

latexmk is a \TeX utility used to monitor changes in source files and recompile as needed.

1. In the tutorial's source code uncomment the latexmk option for the lwarp package:

```
latexmk, % Use latexmk to compile.
```

2. Recompile the printed version of the document.

```
Enter ⇒ lwarpmk print
```

lwarp updates its own configuration files (`lwarpmk.conf` and `tutorial.lwarpmkconf`) whenever the printed version of the document is compiled. These configuration files remember that `lwarpmk` should use `latexmk` to compile the document.

3. Recompile the document.

```
Enter ⇒ lwarpmk print
```

and/or

```
Enter ⇒ lwarpmk html
```

Changes are detected by comparing checksums rather than modification times, so `lwarpmk` again will not trigger a recompile, but `latexmk` has a much better awareness of changes than the `lwarpmk` utility does and it is likely to correctly know when to recompile. A recompile may be forced by making a small change to the source.

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

```
Enter ⇒ lwarpmk print1
```

and/or

```
Enter ⇒ lwarpmk html1
```

5.9 Using XeLaTeX or LuaLaTeX

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

1. Remove the auxiliary files for the project:

```
Enter ⇒ lwarpmk cleanall
```

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

```
Enter ⇒ xelatex tutorial.tex
```

-or-

```
Enter ⇒ lualatex tutorial.tex
```

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

3. To recompile the document:

```
Enter ⇒ lwarpmk print
```

-and-

```
Enter ⇒ lwarpmk html
```

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

5.10 Using a glossary

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

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

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

To process the glossary for the print version:

```
Enter ⇒ lwarpmk printglossary
```

To process the glossary for the HTML version:

```
Enter ⇒ lwarpmk htmlglossary
```

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

5.11 Cleaning auxiliary files

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

```
Enter ⇒ lwarpmk clean
```

5.12 Cleaning auxiliary and output files

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

```
Enter ⇒ lwarpmk cleanall
```

5.13 Processing multiple projects in the same directory

It is possible to have several projects in the same directory. `lwarpmk` has an optional parameter which is the document to compile.

To create each project:

```
Enter ⇒ pdflatex project_a
```

```
Enter ⇒ pdflatex project_b
```

Each project is given its own configuration file:

```
project_a.lwarpmkconf, project_b.lwarpmkconf
```

To compile each project with `lwarkmk`:

```
Enter ⇒ lwarpmk print project_a
```

```
Enter ⇒ lwarpmk html project_b
```

5.14 Using the make utility

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

```
lwarpmk pdftohtml [project]
```

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

6 Additional details

6.1 Font and UTF-8 support

lwarp uses `pdftotext` to convert PDF output into UTF-8-encoded text. This process requires that UTF-8 information be embedded in the PDF file, which usually prevents the use of bit-mapped fonts.

vector fonts
Computer Modern



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

```
usepackage{lmodern}
```

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

```
\usepackage{dejavu}
```

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



`XYTeX` and `LuaTeX` users must use the `fontspec` package. Do NOT use `fontenc`!

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

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

- For `XYTeX` or `LuaTeX`:

Pkg `fontspec`

- `fontspec` and font choices

`ligatures`

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

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

- For `pdflatex`:

Pkg `lmodern`

- `lmodern` or other font-related packages

Pkg `fontenc`

- `fontenc`

Pkg `inputenc`

- `inputenc`

Pkg `newunicodechar`

- `newunicodechar`

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

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

6.1.1 Indexes and UTF-8

lwarp uses the `xindy` program to processes indexes.

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

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

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

6.2 lwarp package loading and options

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

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

Table 4: Package options

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

the math options, see section 7.5.

See table 4 for the full list of options.

6.3 Selecting the operating system

Prog	Unix	lwarp tries to detect which operating system is being used. UNIX / MAC OS / LINUX is the default (collectively referred to as “UNIX” in the configuration files), and MS-WINDOWS is supported as well.
Prog	Mac OS	
Prog	Linux	
Prog	MS-Windows	If WINDOWS is not correctly detected, use the lwarp option OSWindows.
Prog	Windows	
Opt	OSWindows	When detected or specified, the operating-system path separator used by lwarp is modified, the boolean <code>usingOSWindows</code> is set true. This boolean may be tested by the user for later use.

6.4 Selecting actions for print or HTML output

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6.5 Commands to be placed into the warpprint environment

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

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

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

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

6.6 Title page

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

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

The title is used in the meta tags in the HTML files, and the rest are used in `\maketitle`. To use a `\subtitle` or `\published` field, see section 52.7.

<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.

`\title` `{\title}`



Avoid newlines in the `\title`; these will interfere with the file break and CSS detection. Use a `\subtitle` command instead (section 52.7). In HTML, the title will appear in a

heading <h1>.

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

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

`\date` $\{\langle date \rangle\}$

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

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

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

6.7 HTML page meta descriptions

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

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

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

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

6.8 HTML page meta author

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

6.9 CSS

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

File `project.css`

File `sample_project.css`

```

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

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

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

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

```

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

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

```

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

```

6.10 Modifying xindy index processing

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

File `lwarp.xdy`

To customize index processing:

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

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

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

Opt `xdyFilename`

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

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

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

7 Special cases and limitations

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

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

7.1 Formatting

7.1.1 Text formatting

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

7.1.2 Horizontal space

`\hspace` `\hspace` is converted to an inline HTML span of the given width, except that 0 width is ignored, a width of `.16667em` is converted to an HTML thin breakable space (U+2009), and a `\fill` is converted to a `\quad`.

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

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

7.1.3 Text alignment

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

7.1.4 Accents

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

7.1.5 Textcomp

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

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

7.1.6 Superscripts and other non-math uses of math mode

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

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

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

7.1.8 Filenames in lists

filename underscore Escape underscores in the filenames:

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

7.1.9 `resize` package

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

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

While creating SVG math for HTML, the original definitions are temporarily restored, and so should work as expected.

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

7.2 Boxes and minipages

7.2.1 Save Boxes

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

7.2.2 Minipages

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

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

side-by-side Side-by-side minipages may be separated by `\quad`, `\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.

7.2.3 Side-by-side minipages

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

7.2.4 Framed minipages and other environments

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

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

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

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

To frame equations: See section 123 for the fancybox package.

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

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

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

7.2.5 fancybox package

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

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

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

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

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

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

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

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

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

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

7.2.6 mdframed package

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

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

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

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

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

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

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

7.3 Cross-references

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

7.3.1 Page references

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

Ex:

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

`\pagerefPageFor` may be redefined to “page for”, empty, etc. See page 306.

7.3.2 cleveref and varioref packages

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

Ex:

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

See `\cpagerefFor` at page 357 to redefine the message which is printed for page number references.

7.3.3 Hyperlinks, hyperref, and url

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

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

7.3.4 Footnotes and page notes

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

pfnote numbers

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

7.4 Front and back matter

7.4.1 Starred chapters and sections

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

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

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

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

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

⚠ inaccessible HTML page

`\ForceHTMLTOC` To automatically force the HTML version of the document to have a TOC entry for a starred section, use `\ForceHTMLTOC` just before the `\chapter*` or `\section*`. The TOC will only be assigned for HTML output, not for print output, and it will appear in the main TOC and also the sidetoc per page.

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

7.4.2 abstract package

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

7.4.3 titling and authblk

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

⚠ load order

`\published` and `\subtitle` If using the `titling` package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 52.7.

7.4.4 tocloft package

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

⚠ tocloft & other packages

7.4.5 appendix package

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

7.4.6 pagenote package

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

7.4.7 endnotes package

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

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

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

```
\ForceHTMLPage
\theendnotes
```

7.4.8 glossaries package

Pkg glossaries `xindy` is required for glossaries.

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

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

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

Numbered section, on its own HTML page:

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

Unnumbered section, inline with the current HTML page:

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

Unnumbered section, on its own HTML page:

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

- Opt `IndexLanguage` The `lwarp` package takes an option `IndexLanguage=english` to set the language used by `xindy`. This is passed to `xindy` using its `-L` option, and is used for both index and glossary generation.
- Opt `lwarpmk printglossary` `lwarpmk` has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary` to process the glossaries created by `glossaries` using `xindy`.
- Opt `lwarpmk htmlglossary`

7.4.9 Index and the `tocbibind` package

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

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

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

Inline, with a manual TOC entry:

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

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

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

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

Inline, with an automatic TOC entry:

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

```

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

```

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

```

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

```

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

[numbered index section](#)

See section [62](#) for `lwarp`'s core index and glossary code, and section [229](#) for `tocbibind`.

7.5 Math

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

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

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

[SVG image font size](#) The size of the math and text used in the SVG image may be adjusted by setting `\LateximageFontSizeName` to a font size name — *without the backslash*, for ex:

```
\renewcommand{\LateximageFontSizeName}{large}
```

[SVG files](#) As currently implemented, each instance of math creates a new SVG file. In text

¹¹See section [240](#) regarding fonts and fractions.

with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.

SVG inline Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images, but dis-allows the possibility of sharing one file among many instances without user intervention.

PNG files Others have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are also larger as well.

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 8 regarding EPUB output with MathJax.

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

Prog `MathJax`

When MathJax is enabled, math is rendered twice:

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

MathJax limitations Limitations when using MathJax include:

Prog `MathJax`

chapter numbers

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

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

subequations

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

footnotes in math

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

- lateximage
 - Math appearing inside a `lateximage`, and therefore also inside a `Tikz` or `picture` environment, is rendered as SVG math even if MathJax is used in the rest of the document.
- siunitx
 - Usage of `siunitx` inside a math equation is supported via a third-party MathJax extension. While inside a math expression, do not use `\SI` or `\si` inside `\text`, where it will be rendered as normal text.
<https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx>
- ⚠ siunitx inside an equation
 - MathJax does not automatically support custom \LaTeX macros, but they may be set up by the user.
- \LaTeX macros
 - MathJax does not automatically support custom \LaTeX macros, but they may be set up by the user.
- custom MathJax macros For an example of using custom \LaTeX macros with MathJax, see page 327.

7.5.1 ntheorem package

- ⚠ Font control This conversion is not total. Font control is via CSS, and the custom \LaTeX font settings are ignored.
- ⚠ Equation numbering `ntheorem` has a bug with equation numbering in \mathcal{AMS} environments when the option `thref` is used. `lwarp` does not share this bug, so equations with `\split`, etc, are numbered correctly with `lwarp`'s HTML output, but not with the print output. It is recommended to use `cleveref` instead of `ntheorem`'s `thref` option.

7.5.2 siunitx package

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

7.5.3 units and nicefrac packages

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

7.5.4 newtxmath package

- Pkg newtxmath The proper load order is:
- ⚠ loading sequence

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

7.6 Graphics

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

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

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

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

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

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

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

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

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

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

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

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

7.6.1 grffile package

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

7.6.2 color package

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

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

7.6.3 xcolor package

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

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

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

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

tables Colored tables are ignored so far. Use CSS to style tables.

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

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

7.7 Tabular

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

Defining environments:

⚠ misplaced alignment
alignment tab character &

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

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

⚠ tabular inside another
environment

- When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing environment ends, resulting in corrupted HTML rows. To fix this, use \ResumeTabular as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

Cell contents:

⚠ paragraphs

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

⚠ \multirow

- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.
- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so

⚠ `\multicolumn & \multirow`

`\warpprintonly` and `\warpHTMLonly` must be used to make versions for print and HTML.

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

```
\multicolumnrow{2}{c}{c}{3}[0]{1in}[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. Note that recent versions of `multirow` include a new optional `vposn` argument. See section 174.2.

⚠ 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 types
⚠ * column specification

- Vertical rules are not yet supported.
- `*` in a column specification is not used (so far). Repeat the column type the correct number of times.
- Only one each of `@`, `!`, `>`, and `<` may be used at each column, and they are used in that order.
- `\newcolumnntype` is ignored; unknown column types are set to 1.

Rules:

`\multirow` with rules

- If a `multirow` reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. (This may be a browser bug.)

rule at last row

- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.

`\cmidrule` width, trim

- `\cmidrule` does not support width or trim options due to CSS limitations.

`\cmidrule` and
`\multicolumn`

- `\cmidrule` borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has `\multicolumn` cells below `\cmidrule` borders.

⚠ `\warpprintonly`
misplaced `\noalign`

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

```
This & That \endhead
```

to

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

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

Other:

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 \\}
```

7.7.1 longtable package



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

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

instead of

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

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

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



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

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

or place it inside `\warpingprintonly`.

7.7.2 supertabular and xtab packages

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

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

See section 7.7.

⚠ misplaced alignment
alignment tab character &

7.7.3 bigdelim package

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

```

\begin{tabular}{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

7.8 Floats

7.8.1 float, trivfloat, and/or algorithmicx together

- ⚠ **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [232.1](#).

7.8.2 caption and subcaption packages

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

```

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

```

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

7.8.3 subfig package

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

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

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

7.8.4 floatrow package

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

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

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

7.9 Miscellaneous

7.9.1 verse and memoir

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

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

Len `\leftskip` These lengths are used by `verse` and `memoir` to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any `verse` environment, and persist until they

Len `\leftmargini`

Len `\TMLvleftskip`

Len `\TMLleftmargini`

are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

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

7.9.2 newclude package

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

 **loading**

Ex:

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

7.9.3 babel package

Pkg `babel`

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

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

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

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

8 EPUB conversion

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

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

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

Bool `FormatEPUB`

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

To help convert lwarp HTML output to EPUB, add

```
\booltrue{FormatEPUB}
```

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

```
Enter ⇒ lwarpmk cleanall
```

then

```
Enter ⇒ lwarpmk html
```

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

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

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

 **search order**

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

```
Preferences → Plugins → File type plugins → HTML to Zip
```

Check the box Add linked files in breadth first order.

 **section breaks**

The EPUB-conversion program must also know where the section breaks are located. For a list of lwarp's section headings, see table 5. For example, an `article` class document would break at `\section`, which is mapped to HTML heading level `<h4>`, whereas a `book` class document would break at `\chapter`, which is HTML heading level `<h3>`. For Calibre, this option is found in

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

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

Match HTML tags with tag name:

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

`//h:h4` — or — `//h:h3`

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

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

MathJax support

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

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

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

9 Word-processor conversion

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

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

Bool `FormatWordProcessor`

Default false. Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments.

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

```
\booltrue{FormatWordProcessor}
```

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

- Headers, footers, and navigation are removed at file splits.
- Any accumulated footnotes are printed at the bottom of each file split. These will have to be manually moved to their proper place in the document. lwarp does not know where the page breaks will be in the word processor's document, so the footnotes are simply moved to the end of each sectional break.
- Forces single-file output.
- Turns off HTML debugging comments. These are comments appearing inside the HTML code, marking the opening/closing of sections and `<div>`s, but they are no longer useful when the document has been imported into a word processor.

An additional boolean may be set to help mark float boundaries:

Default true. Adds

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

or

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

Bool HTMLMarkFloats

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

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

10 Modifying lwarp

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

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

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

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

10.1 Creating an lwarp version of a package

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

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

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

Each `lwarp-` package should first call either

```
\LWR@ProvidesPackageDrop
```

or

```
\LWR@ProvidesPackagePass
```

If `Dropped`, the original print-version package is ignored, and only the `lwarp-` version is used. Use this where the original print version is useless for HTML. If `Passed`, the original package is loaded first, with the user-supplied options, then the `lwarp-` version continues loading as well. See section 183 ([ntheorem](#)) for an example of selectively disabling user options for a package. Use this when HTML output only requires some modifications of the original package. For a case where the original package is usable without changes, there is no need to create a `lwarp-` version.

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

When adding a package to `lwarp.dtx` for permanent including in `lwarp`, provide the `lwarp-<packagename>` code in `lwarp.dtx`, add its entry into `lwarp.ins`, and also remember to add

```
\LWR@loadafter{<packagename>}
```

to `lwarp.dtx` in section 20. This causes `lwarp` to stop with an error if `packagename` is loaded before `lwarp`.

10.3 Testing `lwarp`

When changes have been made, test the print output before testing the HTML. The print output compiles faster, and any errors in the printed version will be easier to figure out than the HTML version.

Remember that the configuration files are only rewritten when compiling the printed version of the document.

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

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

10.4 Modifying `lwarpmk`

Prog `lwarpmk`
File `lwarpmk.lua`

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

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

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

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

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

11 Troubleshooting

11.1 Using the `lwarp.sty` package

Also see:

Section 6.5: [Commands to be placed into the `warpprint` environment](#)

Section 7: [Special cases and limitations](#)

Section 7: [Special cases and limitations](#)

Text is not converting:

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

Undefined HTML settings:

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

Tabular problems: See section 7.7.

Obscure error messages:

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

Missing sections:

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

Missing HTML files:

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

Missing / incorrect cross-references:

- Use `lwarpmk` again followed by `lwarpmk html` or `lwarpmk print` to compile the document one more time.
- `\nameref` refers to the most recently-used section where the `\label` was defined. If no section has been defined before the `\label`, the link will be empty. Index entries also use `\nameref` and have the same limitation.
- `cleveref` and `varioref` are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to “for”.

 [empty link](#)

 [cleveref page numbers](#)

Ex:
`\cpageref{tab:first,tab:second}`
 in HTML becomes:
 “pages **for** table 4.1 and **for** table 4.2”
 See `\cpagerefFor` at page 357 to redefine the message which is printed
 for page number references.

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

Use \XeTeX or \LuaTeX .

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}` in-
 stead of `[warpHTML]`.

HTML document contains a single unformatted print document:

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

Images are appearing in strange places:

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

⚠ Adding/removing

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

⚠ HTML instead of images

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

⚠ Lots of files!

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

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

Plain-looking document:

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

Broken fragments of HTML:

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

Changes do not seem to be taking effect:

- Be sure to `lwarpmk clean`, recompile, then start by reloading the home page. You may have been looking at an older version of the document. If you changed a section name, you may have been looking at the file for the old name.
- See the warning regarding changes to the HTML settings at section 5.7.
- Verify that the proper CSS is actually being used.
- The browser may compensate for some subtle changes, such as automatically generating ligatures, reflowing text, etc.

Un-matched conditional compiles:

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

11.1.1 Debug tracing output

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

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

11.2 Compiling the `lwarp.dtx` file

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

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

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

Un-nested environments:

Be sure to properly nest:

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

12 Implementation

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

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

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

13 Stack depths

Stacks are created to track depth inside the \TeX document structure. This depth is translated to HTML headings as shown in table 5. “Depth” here is not depth in the traditional computer-science stack-usage sense, but rather a representation of the nesting depth inside the \TeX document structure.

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

Table 5: Section depths and HTML headings

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

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

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

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

14 Source Code

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

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

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

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

objects Black-colored tags in the left margin are used to identify programming objects such as files, packages, environments, booleans, and counters. Items without a tag are

index entries command macros. Each of these also appears in the index as individual entries, and are also listed together under “files”, “packages”, “environments”, “booleans”, and “counters”.

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

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

15 Detecting the T_EX Engine — pdf_latex, lua_latex, xe_latex

```

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

```

16 Unicode Input Characters

for HTML & PRINT:

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

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

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

```

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

```

In PDF_T_EX, preserve upright quotes in verbatim text:

```

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

```

17 Early package requirements

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

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

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

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

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

```
25 \RequirePackage{letltxmacro}
```

18 Operating-System portability

Prog	Unix	lwarp tries to detect which operating system is being used. UNIX / MAC OS / LINUX is the default (collectively referred to as “UNIX” in the configuration files), and MS-WINDOWS is supported as well.
Prog	Mac OS	
Prog	Linux	
Prog	MS-Windows	If WINDOWS is not correctly detected, use the lwarp option OSWindows.
Prog	Windows	When detected or specified, the operating-system path separator used by lwarp is modified, the boolean usingOSWindows is set true. This boolean may be tested by the user for later use.
Opt	OSWindows	

18.1 Common portability code

Bool usingOSWindows Set if the OSWindows option is used.

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

18.2 Unix, Linux, and Mac OS

\OSPathSymbol Symbol used to separate directories in a path.

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

18.3 MS-Windows

For MS-Windows:

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

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

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

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

19 Package options

Pkg `kvoptions` Allows key/value package options.

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

Bool `warpingprint`

Bool `warpingHTML`

Bool `mathjax`

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

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

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

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

Opt `warpprint` If the `warpprint` option is given, boolean `warpingprint` is true and boolean `warpingHTML` is false, and may be used for `\ifbool` tests.

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

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

Opt `warpHTML` If the `warpHTML` option is given, boolean `warpingHTML` is true and boolean `warpingprint` is false, and may be used for `\ifbool` tests.

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

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

Opt `mathsvg` Option `mathsvg` selects SVG math display: If the `mathsvg` option is given, boolean `mathjax` is false, and may be used for `\ifbool` tests.

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

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

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

Opt `BaseJobname` Option `BaseJobname` sets the `\BaseJobname` for this document.

This is the `\jobname` of the printed version, even if currently compiling the HTML version. I.e. this is the `\jobname` without `_html` appended. This is used to set `\HomeHTMLFilename` if the user did not provide one.

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

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

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

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

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

```
65 \DeclareStringOption[lwarp.xdy]{xdyFilename}
```

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

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

```

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

```

Opt OSWindows Tells lwarp to use MS-Windows compatibility. Auto-detection of the operating system is attempted, and this option is only necessary if the auto-detection fails. See the automatically-generated `lwarpmk.conf` file to find out whether the operating system was detected correctly.

```

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

```

Opt HomeHTMLFilename The filename of the homepage. The default is the jobname. This option is stored into `\LWR@HomeHTMLFilename`, and later transferred into `\HomeHTMLFilename` for internal use.

```

77 \DeclareStringOption[] {HomeHTMLFilename}

```

Opt HTMLFilename The filename prefix of web pages after the homepage. The default is empty, no prefix. This option is stored into `\LWR@HTMLFilename`, and later transferred into `\HTMLFilename` for internal use.

```

78 \DeclareStringOption[] {HTMLFilename}

```

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

```

79 \DeclareBoolOption[false]{latexmk}

```

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

```

80 \ProcessKeyvalOptions*\relax

```

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

```

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

```

Defaults unless already over-ridden by the user:

```

82 \ifcsemt{LWR@HomeHTMLFilename}{
83 \newcommand*{\HomeHTMLFilename}{\BaseJobname}

```

```

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

```

19.1 Conditional compilation

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

Only process the contents if producing printed output.

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

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

Only process the contents if producing HTML output.

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

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

```
91 \RequirePackage{comment}
```

Use `comment_print.cut` for print mode, and `comment_html.cut` for HTML mode. This helps `latexmk` to more reliably know whether to recompile.

```

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

```

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

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

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

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

`\Env` `warpHTML`

For HTML output:

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

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

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

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

20 Misplaced packages

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

Packages which should only be loaded before lwarp have their own

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

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

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

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

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

```

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

```

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

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

```

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

```

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

The following packages must be loaded after lwarp:

```

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

```

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

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

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

21 Required packages

These packages are automatically loaded by lwarp when generating HTML output. Some of them are also automatically loaded when generating print output, but some are not.

In the document preamble, create a `warpprint` environment, and place inside it any of the following packages which are required and which are labeled as “Print: OK

to Load in a warpprint environment”. Those packages which are labeled as “Print: Pre-Loaded” need not be placed into the document preamble.

for HTML & PRINT: 296 `\begin{warppall}`

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

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

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

308 \end{warppall}
```

for HTML output: 309 `\begin{warppHTML}`

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

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

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

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

Pkg microtype

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

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

```

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

```

Pkg geometry Tactics to avoid unwanted page breaks and margin overflow:

- Uses a very long and wide page to minimize page breaks and margin overflow.
- Uses a scriptsize font.
- Uses extra space at the margin to avoid HTML tag overflow off the page.
- Forces a new PDF page before some environments.
- Forces line break between major pieces of long tags.

for HTML output:

```

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

```

for HTML & PRINT: 336 \begin{warpall}

Pkg xparse

LaTeX3 command argument parsing

```

337 \RequirePackage{xparse}

```

Pkg afterpackage Used to patch titling to add \AddSubtitlePublished.

```

338 \RequirePackage{afterpackage}

```

```

339 \end{warpall}

```

for HTML output: 340 \begin{warpHTML}

Pkg `expl3`

L^AT_EX3 programming

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

Pkg `getttitlestring`

Used to emulate `\nameref`.

```
342 \RequirePackage{getttitlestring}
```

Pkg `everyhook`

`everyhook` is used to patch paragraph handling.

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

for HTML & PRINT:

```
345 \begin{warpall}
```

Pkg `filecontents`

Used to write helper files, done in print mode.

Patched to work with `morewrites`, per <https://tex.stackexchange.com/questions/312830/does-morewrites-not-support-filecontents-and-can-i-write-body-of-environment-us/312910>

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

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

for HTML output:

```
356 \begin{warpHTML}
```

Pkg `xifthen`

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

Pkg `xstring`

```
358 \RequirePackage{xstring}
```

Pkg xstring

```
359 \RequirePackage{verbatim}
```

Pkg makeidx

```
360 \RequirePackage{makeidx}
```

```
361 \makeindex
```

Pkg calc

```
362 \RequirePackage{calc}
```

Pkg refcount

```
363 \RequirePackage{refcount}
```

Pkg newfloat

```
364 \RequirePackage{newfloat}
```

Pkg caption

```
365 \RequirePackage{caption}
```

```
366 \end{warpHTML}
```

for HTML & PRINT: 367 \begin{warpall}

Pkg environ

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

```
368 \RequirePackage{environ}
```

```
369 \end{warpall}
```

for HTML output: 370 \begin{warpHTML}

Pkg zref

Used for cross-references.

```
371 \RequirePackage{zref}
```

Pkg amsmath

Equation numbers are placed to the left for HTML.

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

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

Used to convert lengths for image width/height options.

```
374 \RequirePackage{printlen}

375 \end{warpHTML}
```

22 Loading packages

for HTML output: 376 `\begin{warpHTML}`

Remember the original `\RequirePackage`:

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

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

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

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

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

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

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

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

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

```
383 \LWR@findword{\LWR@requirepackagenames}{#1}[\LWR@strresult]%
```

Remove blanks. The original name with blanks is in LWR@strresult and the final name with no blanks goes into LWR@strresulttwo.

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

See if the package name was found:

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

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

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

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

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

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

Redirect up to nine names:

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

`\RequirePackage` depending on the options and version:

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

```

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

```

`\LWR@ProvidesPackagePass` $\langle pkgname \rangle$ [$\langle version \rangle$]

Uses the original package, including options.

```

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

```

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

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

```

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

442 \end{warpHTML}

```

23 File handles

Defines file handles for writes.

for HTML & PRINT: 443 `\begin{warpall}`

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

444 `\newwrite\LWR@quickfile%`

445 `\end{warpall}`

for HTML output: 446 `\begin{warpHTML}`

`\LWR@lateximagesfile` For lateximages.txt.

447 `\newwrite\LWR@lateximagesfile`

448 `\end{warpHTML}`

24 Include a file

During HTML output, `\include{<filename>}` causes the following to occur:

1. lwarp creates `<filename>_html_inc.tex` whose contents are:


```
\input <filename>.tex
```
2. `<filename>_html_inc.tex` is then `\included` instead of `<filename>.tex`.
3. `<filename>_html_inc.aux` is automatically generated and used by \LaTeX .

for HTML output: 449 `\begin{warpHTML}`

`\include` `{<filename>}`

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

450 `\def\@include#1 {%`

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

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

```

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

483 \end{warpHTML}

```

25 Copying a file

for HTML output: 484 \begin{warpHTML}

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

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

```

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

```

```

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

502 \end{warpHTML}

```

26 Debugging messages

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

Bool LWR@tracinglwarp True if tracing is turned on.

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

\tracinglwarp Turns on the debug tracing messages.

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

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

```

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

```

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

```

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

```

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

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

27 HTML-conversion output modifications

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

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

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

`Bool FormatWordProcessor` Default false. Changes HTML output for easier conversion by a word processor. Removes headers and nav, prints footnotes per section, and also forces single-file output and turns off HTML debug comments.

```
525 \newbool{FormatWordProcessor}
526 \boolfalse{FormatWordProcessor}
```

`Bool HTMLMarkFloats` Default true. Adds

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

or

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

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

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

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

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

28 Remembering original formatting macros

for HTML output: 530 \begin{warpHTML}

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

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

```

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

```

29 Accents

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

for HTML output: `596 \begin{warpHTML}`

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

```
597 \AtBeginDocument{
```

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

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

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

For pdf \LaTeX , additional work is required:

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

The HTML redefinitions follow.

For pdf \LaTeX , Unicode diacritical marks are used:

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

For all engines, a Unicode diacritical tie is used:

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

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

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

```
633 }% AtBeginDocument
```

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

30 Configuration Files

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

30.1 project_html.tex

File `project_html.tex` Used to allow an HTML version of the document to exist alongside the print version.

Only write `\jobname_html.tex` if generating the print version.

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

30.2 lwarpmk.conf

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

An example `lwarpmk.conf`:

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

for PRINT output:

```

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

```

30.3 project.lwarpmkconf

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

```

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

```

```

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

```

30.4 lwarp.css

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

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

```

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
1628 div.published
1629 {
1630     text-align: center ;
1631     font-variant: normal ;
1632     font-style: italic ;
1633     font-size: 1em ;
1634     margin: 3ex 0em 3ex 0em ;
1635 }
1636
1637 div.subtitle
1638 {
1639     text-align: center ;
1640     font-variant: normal ;
1641     font-style: italic ;
1642     font-size: 1.25em ;
1643     margin: 3ex 0em 3ex 0em ;
1644 }
1645
1646 div.subtitle p { margin: 1ex ; }
1647
1648 div.author
1649 {
1650     font-variant: normal ;
1651     font-style: normal ;
1652     font-size: 1em ;
1653     margin: 3ex 0em 3ex 0em ;
1654 }
1655
1656 div.oneauthor {
1657     display: inline-block ;
1658     margin: 3ex 1em 0ex 1em ;
1659 }
1660
1661 /*
1662 div.author table {
1663     margin: 3ex auto 0ex auto ;
1664     background: none ;
1665 }
1666
1667 div.author table tbody tr td { padding: .25ex ; }
1668 */
1669
1670 span.affiliation {font-size: .85em ; font-variant: small-caps; }
1671
1672 div.titledate {
1673     text-align: center ;
1674     font-size: .85em ;
1675     font-style: italic;
1676     margin: 6ex 0em 6ex 0em ;
1677 }
```

```
1678
1679
1680 nav.topnavigation{
1681     text-align: left ;
1682     padding: 0.5ex 1em 0.5ex 1em ;
1683 /*     margin: 2ex 0em 3ex 0em ; */
1684     margin: 0 ;
1685     border-bottom: 1px solid silver ;
1686     border-top: 1px solid silver ;
1687     clear:right ;
1688 }
1689
1690 nav.botnavigation{
1691     text-align: left ;
1692     padding: 0.5ex 1em 0.5ex 1em ;
1693 /*     margin: 3ex 0em 2ex 0em ; */
1694     margin: 0 ;
1695     border-top: 1px solid silver ;
1696     border-bottom: 1px solid silver ;
1697     clear:right ;
1698 }
1699
1700
1701 header{
1702     line-height: 1.2 ;
1703     font-size: 1em ;
1704 /*     border-bottom: 2px solid silver ; */
1705     margin: 0px ;
1706     padding: 0ex 1em 0ex 1em ;
1707     text-align:center ;
1708 }
1709
1710 header p {margin:0ex;padding:4ex 0em 2ex 0em ;text-align:center;}
1711
1712
1713 footer{
1714     font-size: .85em ;
1715     line-height: 1.2 ;
1716     margin-top: 1ex ;
1717     border-top: 2px solid silver ;
1718     padding: 2ex 1em 2ex 1em ;
1719     clear:right ;
1720     text-align:left ;
1721 }
1722
1723
1724 a.linkhome { font-weight:bold ; font-size: 1em ;}
1725
1726
1727 div.lateximagesource { padding: 0px ; margin: 0px ; display: none; }
```

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

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

```
1828
1829
1830 span.amsthmnoteplain,
1831 span.amsthmnotedefinition
1832 {font-style:normal ;}
1833
1834
1835 span.theoremheaderremark,
1836 span.theoremheaderproof,
1837 span.amsthmproofname
1838 {font-style:italic ; font-weight: normal ; margin-right: 1em ; }
1839
1840 span.theoremheadersc
1841 {
1842     font-style:normal ;
1843     font-variant: small-caps ;
1844     font-weight: normal ;
1845     margin-right: 1em ;
1846 }
1847
1848 .theoremdemark {float:right}
1849
1850 div.amsthmbodyplain, div.theorembodyplain, div.theorembodynonumberplain,
1851 div.theorembodybreak, div.theorembodynonumberbreak,
1852 div.theorembodymarginbreak,
1853 div.theorembodychangebreak,
1854 div.theorembodychange,
1855 div.theorembodymargin
1856 {
1857     font-style:italic;
1858     margin-top: 3ex ; margin-bottom: 3ex ;
1859 }
1860
1861 div.theorembodydefinition, div.theorembodyremark, div.theorembodyproof,
1862 div.theorembodyplainupright, nonumberplainuprightsc,
1863 div.amsthmbodydefinition, div.amsthmbodyremark,
1864 div.amsthmproof
1865 {
1866     font-style: normal ;
1867     margin-top: 3ex ; margin-bottom: 3ex ;
1868 }
1869
1870 span.amsthmnoteremark {}
1871
1872
1873
1874 /*
1875 For CSS LaTeX and related logos:
1876 Based on:
1877 http://edward.oconnor.cx/2007/08/tex-poshlet
```

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

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

```

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

```

30.5 lwarp_sagebrush.css

File `lwarp_sagebrush.css` An optional CSS which may be used for a semi-modern appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```

2008 \begin{warpprint}
2009 \begin{filecontents*}{lwarp_sagebrush.css}
2010 @import url("lwarp.css") ;
2011
2012
2013 A:link {color:#105030 ; text-decoration: none ; }
2014 A:visited {color:#705030 ; text-shadow:1px 1px 2px #a0a0a0;}
2015 A:hover {color:#006000 ; text-decoration: underline ; text-shadow:0px 0px 2px #a0a0a0;}
2016 A:active {color:#00C000 ; text-shadow:1px 1px 2px #a0a0a0;}

```

```
2017
2018
2019
2020 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2021 {
2022     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2023         "Linux Libertine 0", "Liberation Serif",
2024         "Nimbus Roman No 9 L", "FreeSerif",
2025         "Hoefler Text", Times, "Times New Roman", serif;
2026     font-variant: small-caps ;
2027     font-weight: normal ;
2028     color: #304070 ;
2029     text-shadow: 2px 2px 3px #808080;
2030 }
2031
2032 h1 { /* title of the entire website, used on each page */
2033     font-variant: small-caps ;
2034     color: #304070 ;
2035     text-shadow: 2px 2px 3px #808080;
2036     background-color: #F7F7F0 ;
2037     background-image: linear-gradient(to bottom, #F7F7F0, #C0C0C4);
2038 }
2039
2040 h1 {
2041     border-bottom: 1px solid #304070;
2042     border-top: 2px solid #304070;
2043 }
2044
2045 h2 {
2046     border-bottom: 1px solid #304070;
2047     border-top: 2px solid #304070;
2048     background-color: #F7F7F0 ;
2049     background-image: linear-gradient(to bottom, #F7F7F0, #DAD0C0);
2050 }
2051
2052
2053
2054 div.abstract {
2055     background: #f5f5eb ;
2056     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2057
2058     border: 1px solid silver;
2059     border-radius: 1em ;
2060 }
2061
2062 div.abstract dl {line-height:1.5;}
2063 div.abstract dt {color:#304070;}
2064
2065 div.abstracttitle{
2066     font-family: "URW Classico", Optima, "Linux Biolinum 0",
```

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

```
2117     background-image: linear-gradient(to bottom, #f5f5eb, #C8C8B8);
2118
2119 }
2120
2121 div.sidebartitle{
2122     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2123         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2124         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2125     font-weight:bold;
2126     font-variant: small-caps ;
2127     border-bottom: 1px solid silver ;
2128     color: #304070 ;
2129     text-align: center ;
2130     text-shadow: 1px 1px 2px #808080;
2131 }
2132
2133
2134 .fancyvrblabel {
2135     font-family: "URW Classico", Optima, "Linux Biolinum 0",
2136         "Linux Libertine 0", "Liberation Serif", "Nimbus Roman No 9 L",
2137         "FreeSerif", "Hoefler Text", Times, "Times New Roman", serif;
2138     font-weight:bold;
2139     font-variant: small-caps ;
2140     font-size: 1.5em ;
2141     color: #304070 ;
2142     text-align: center ;
2143     text-shadow: 1px 1px 2px #808080;
2144 }
2145
2146
2147
2148 div.minipage {
2149     background-color: #eeeeee7 ;
2150     border: 1px solid silver ;
2151     border-radius: 1ex ;
2152 }
2153
2154 section.textbody > div.minipage {
2155     box-shadow: 3px 3px 3px #808080 ;
2156 }
2157
2158 div.fboxBlock div.minipage { box-shadow: none ; }
2159
2160 .framed .minipage , .framedleftbar .minipage {
2161     border: none ;
2162     background: none ;
2163     padding: 0ex ;
2164     margin: 0ex ;
2165 }
2166
```

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

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

30.6 lwarp_formal.css

File `lwarp_formal.css` An optional css which may be used for a more formal appearance.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```
2253 \begin{warpprint}
2254 \begin{filecontents*}{lwarp_formal.css}
2255 @import url("lwarp.css") ;
```

```
2256
2257
2258
2259 A:link {color:#802020 ; text-decoration:none; }
2260 A:visited {color:#802020 ; text-shadow:none ;}
2261 A:hover {color:#400000 ; text-shadow:none ;}
2262 A:active {color:#C00000 ; text-shadow:none ;}
2263
2264
2265 body {
2266     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2267         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2268         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2269         "Times New Roman", serif;
2270     background: #fffcf5;
2271 }
2272
2273 span.textrm {
2274     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2275         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2276         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2277         "Times New Roman", serif;
2278 }
2279
2280 span.textsf {
2281     font-family: "DejaVu Sans", "Bitstream Vera Sans",
2282         Geneva, Verdana, sans-serif ;
2283 }
2284
2285
2286
2287 h1, h2, h3, h4, h5, h6, span.paragraph, span.subparagraph
2288 {
2289     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2290         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2291         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2292         "Times New Roman", serif;
2293     color: #800000 ;
2294     text-shadow: none ;
2295 }
2296
2297 h1, h2 {
2298     background-color: #fffcf5 ;
2299     background-image: none ;
2300     border-bottom: 1px solid #808080;
2301     border-top: 2px solid #808080;
2302 }
2303
2304 div.abstracttitle {
2305     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
```

```
2306     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2307     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2308     "Times New Roman", serif;
2309     color: black ;
2310     text-shadow: none ;
2311 }
2312
2313 span.abstractrunintitle {
2314     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2315     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2316     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2317     "Times New Roman", serif;
2318     color: black ;
2319     text-shadow: none ;
2320 }
2321
2322 div.abstract { font-size: 100% }
2323
2324 .sidebar {
2325     background: #fffcf5;
2326     background-image: none ;
2327     margin: 2em 5% 2em 5%;
2328     padding: 0.5em 1em;
2329     border: none ;
2330     border-top : 1px solid silver;
2331     border-bottom : 1px solid silver;
2332     font-size: 90% ;
2333 }
2334
2335 div.sidebar{
2336     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2337     "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2338     "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2339     "Times New Roman", serif;
2340     color: #800000 ;
2341     text-shadow: none ;
2342     border: none ;
2343 }
2344
2345 .example {
2346     background: #fffcf5;
2347     background-image: none ;
2348     margin: 2em 5% 2em 5%;
2349     padding: 0.5em 1em;
2350     border: none ;
2351     border-top : 1px solid silver;
2352     border-bottom : 1px solid silver;
2353 }
2354
2355 div.exampletitle{
```

```
2356 font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2357 "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2358 "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2359 "Times New Roman", serif;
2360 color: #800000 ;
2361 text-shadow: none ;
2362 border: none ;
2363 }
2364
2365 div.fancyvrblabel{
2366 font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2367 "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2368 "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2369 "Times New Roman", serif;
2370 color: #800000 ;
2371 text-shadow: none ;
2372 border: none ;
2373 }
2374
2375
2376
2377 .verse {
2378 font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2379 "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2380 "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2381 "Times New Roman", serif;
2382 }
2383
2384
2385 figure {
2386 margin: 3ex 5% 3ex 5% ;
2387 padding: 1ex 1em 1ex 1em ;
2388 background-color: #fffcf5 ;
2389 overflow-x: auto ;
2390 border: none ;
2391 /* border-top: 1px solid silver; */
2392 /* border-bottom: 1px solid silver; */
2393 }
2394
2395
2396 figcaption , .lstlisting {
2397 border: none ;
2398 /* border-top: 1px solid silver ; */
2399 /* border-bottom: 1px solid silver ; */
2400 background-color: #fffcf5 ;
2401 }
2402
2403 .tnotes {
2404 background: #fffcf5 ;
2405 }
```

```
2406
2407 .theorem {
2408     background: none ;
2409 }
2410
2411 .minipage {
2412     background-color: #fffcf5 ;
2413     border: none ;
2414 }
2415
2416 div.floatrow figure { border: none ; }
2417
2418 figure figure { border: none ; }
2419
2420
2421 nav.toc, nav.lof, nav.lot, nav.lol {
2422     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2423         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2424         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2425         "Times New Roman", serif;
2426 }
2427
2428 nav.sidetoc {
2429     font-family: "Linux Libertine O", "Hoefler Text", "Garamond",
2430         "Bembo", "Janson", "TeX Gyre Pagella", "Palatino",
2431         "Liberation Serif", "Nimbus Roman No 9 L", "FreeSerif", Times,
2432         "Times New Roman", serif;
2433     background-image: linear-gradient(to bottom, #fffcf5, #COCOC0);
2434     border-radius: 0px 0px 0px 20px ;
2435 }
2436
2437 div.sidetoctitle{
2438     color: #800000 ;
2439 }
2440
2441 header{
2442     background-color: #e0e0e0 ;
2443     background-image: linear-gradient(to top, #fffcf5, #b0b0b0);
2444     text-align:center ;
2445 }
2446
2447 footer{
2448     background-color: #e0e0e0 ;
2449     background-image: linear-gradient(to bottom, #fffcf5, #b0b0b0);
2450     padding: 2ex 1em 2ex 1em ;
2451     clear:right ;
2452     text-align:left ;
2453 }
2454
2455 nav.botnavigation {
```

```

2456     background: #dedcd5 ;
2457     border-top: 1px solid black ;
2458 }
2459 \end{filecontents*}
2460 % \end{Verbatim}% for syntax highlighting
2461 \end{warpprint}

```

30.7 sample_project.css

File `sample_project.css` The project-specific css file. Use with `\CSSFilename`.

If used, this must be present both when compiling the project and also when distributing the HTML files.

```

2462 \begin{warpprint}
2463 \begin{filecontents*}{sample_project.css}
2464 /* ( --- Start of project.css --- ) */
2465 /* A sample project-specific CSS file for lwarp --- ) */
2466
2467 /* Load default lwarp settings: */
2468 @import url("lwarp.css") ;
2469 /* or lwarp_formal.css, lwarp_sagebrush.css */
2470
2471 /* Project-specific CSS setting follow here. */
2472 /* . . . */
2473
2474 /* ( --- End of project.css --- ) */
2475 \end{filecontents*}
2476 % \end{Verbatim}% for syntax highlighting
2477 \end{warpprint}

```

30.8 lwarp.xdy

File `lwarp.xdy` Used to modify the index for lwarp.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```

2478 \begin{warpprint}
2479 \begin{filecontents*}{lwarp.xdy}
2480 (require "tex/inputenc/latin.xdy")
2481 (merge-rule "\\PS *" "Postscript")
2482 (require "texindy.xdy")
2483 (require "page-ranges.xdy")
2484 (require "book-order.xdy")

```

```

2485 (markup-locref :open "\hyperindexref{" :close "}")
2486 \end{filecontents*}
2487 % \end{Verbatim}% for syntax highlighting
2488 \end{warpprint}

```

30.9 lwarp_mathjax.txt

File `lwarp_mathjax.txt` Used by lwarp when using MathJax.

This must be present when compiling the project, but does not need to be present when distributing the resulting HTML files.

```

2489 \begin{warpprint}
2490 \begin{filecontents*}{lwarp_mathjax.txt}
2491 <!-- https://groups.google.com/forum/#!topic/
2492                               mathjax-users/jUtewUcE2bY -->
2493 <script type="text/x-mathjax-config">
2494 MathJax.Hub.Register.StartupHook("TeX AMSmath Ready",function () {
2495   var seteqsectionDefault = {name: "", num: 0};
2496   var seteqsections = {}, seteqsection = seteqsectionDefault;
2497   var TEX = MathJax.InputJax.TeX, PARSE = TEX.Parse;
2498   var AMS = MathJax.Extension["TeX/AMSmath"];
2499   TEX.Definitions.Add({
2500     macros: {
2501       seteqsection: "mySection",
2502       seteqnumber: "mySetEqNumber"
2503     }
2504   });
2505
2506   PARSE.Augment({
2507     mySection: function (name) {
2508       seteqsection.num = AMS.number;
2509       var n = this.GetArgument(name);
2510       if (n === "") {
2511         seteqsection = seteqsectionDefault;
2512       } else {
2513         if (!seteqsections["_" + n])
2514           seteqsections["_" + n] = {name:n, num:0};
2515         seteqsection = seteqsections["_" + n];
2516       }
2517       AMS.number = seteqsection.num;
2518     },
2519     mySetEqNumber: function (name) {
2520       var n = this.GetArgument(name);
2521       if (!n || !n.match(/^\s*[0-9]+\s*$/))
2522         n = ""; else n = parseInt(n)-1;
2523       <!-- $ syntax highlighting -->

```

```
2524     if (n === "" || n < 1)
2525         TEX.Error
2526             ("Argument to "+name+" should be a positive integer");
2527     AMS.number = n;
2528 }
2529 });
2530 MathJax.Hub.Config({
2531   TeX: {
2532     equationNumbers: {
2533       formatTag: function (n)
2534         {return "("+(seteqsection.name+"."+n).replace(/\^\.\/,"")+"}"},
2535       formatID: function (n) {
2536         n = (seteqsection.name+'.'+n).replace
2537             (/[:'<>&]/g,"").replace(/\^\.\/,"");
2538         return 'mjax-eqn-' + n;
2539       }
2540     }
2541   }
2542 });
2543 });
2544 </script>
2545
2546 <!-- http://docs.mathjax.org/en/latest/options/ThirdParty.html -->
2547 <script type="text/x-mathjax-config">
2548   MathJax.Ajax.config.path["Contrib"] =
2549     "https://cdn.mathjax.org/mathjax/contrib";
2550 </script>
2551
2552 <!-- https://github.com/mathjax/MathJax-third-party-extensions/
2553         tree/master/siunitx -->
2554 <script type="text/x-mathjax-config">
2555   MathJax.Hub.Config({
2556     extensions: ["tex2jax.js","[Contrib]/siunitx/siunitx.js"],
2557     jax: ["input/TeX","output/HTML-CSS"],
2558     tex2jax: {inlineMath: [[["$","$"],["\\(","\\)"]]},
2559     TeX: {extensions: ["AMSmath.js","AMSsymbols.js", "sinunitx.js"]}
2560   });
2561 </script>
2562
2563 <script type="text/x-mathjax-config">
2564   MathJax.Hub.Config({
2565     TeX: {
2566       equationNumbers: {
2567         autoNumber: "AMS"
2568       }
2569     }
2570   });
2571 </script>
2572
2573 <!-- Alternative CDN provider: -->
```

```

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

```

30.10 lwarpmk option

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

```
2588 \begin{LWR@createlwarpmk}
```

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

```

2589 \begin{filecontents*}{lwarpmk.lua}
2590 #!/usr/bin/env texlua
2591
2592 -- Copyright 2016-2017 Brian Dunn
2593
2594 -- Print the usage of the lwarpmk command:
2595
2596 printversion = "v0.40"
2597
2598 function printhelp ()
2599 print ("lwarpmk: Use lwarpmk -h or lwarpmk --help for help.") ;
2600 end
2601
2602 function printusage ()
2603 print ( [[
2604
2605 lwarpmk print [project]: Compile the print version if necessary.
2606 lwarpmk print1 [project]: Forced single compile of the print version.
2607 lwarpmk printindex [project]: Process the index for the print version.
2608 lwarpmk printglossary [project]: Process the glossary for the print version.
2609 lwarpmk html [project]: Compile the HTML version if necessary.
2610 lwarpmk html1 [project]: Forced single compile of the HTML version.
2611 lwarpmk htmlindex [project]: Process the index for the html version.
2612 lwarpmk htmlglossary [project]: Process the glossary for the html version.

```

```
2613 lwarpmk again [project]: Touch the source code to trigger recompiles.
2614 lwarpmk limages [project]: Process the "lateximages" created by lwarp.sty.
2615 lwarpmk pdftohtml [project]:
2616     For use with latexmk or a Makefile:
2617     Convert project_html.pdf to project_html.html and
2618     individual HTML files.
2619 lwarpmk clean [project]: Remove project.aux, .toc, .lof/t, .idx, .ind, .log, *_html_inc.*, .gl*
2620 lwarpmk cleanall [project]: Remove auxiliary files and also project.pdf, *.html
2621 lwarpmk -h: Print this help message.
2622 lwarpmk --help: Print this help message.
2623
2624 ]] )
2625 printconf ()
2626 end
2627
2628 -- Print the format of the configuration file lwarpmk.conf:
2629
2630 function printconf ()
2631 print ( [[
2632 An example lwarpmk.conf or <project>.lwarpmkconf project file:
2633 --
2634 opsystem = "Unix"    (or "Windows")
2635 latexname = "pdflatex" (or "lualatex", or "xelatex")
2636 sourcename = "projectname" (the source-code filename w/o .tex)
2637 homehtmlfilename = "index" (or perhaps the project name)
2638 htmlfilename = "" (or "projectname" - filename prefix)
2639 latexmk = "false" (or "true" to use latexmk to build PDFs)
2640 languge = "english" (use a language supported by xindy)
2641 xdyfile = "lwarp.xdy" (or a custom file based on lwarp.xdy)
2642 --
2643 Filenames must contain only letters, numbers, underscore, or dash.
2644 Values must be in "quotes".
2645
2646 ]] ) ;
2647 end
2648
2649
2650 -- Split one large sourcefile into a number of files,
2651 -- starting with destfile.
2652 -- The file is split at each occurrence of <!--|Start file|newfilename|*
2653
2654 function splitfile (destfile,sourcefile)
2655 print ("lwarpmk: Splitting " .. sourcefile .. " into " .. destfile) ;
2656 local sfile = io.open(sourcefile)
2657 io.output(destfile)
2658 for line in sfile:lines() do
2659 i,j,copen,cstart,newfilename = string.find (line,"(.*)|(.*)|(.*)|") ;
2660 if ( (i~= nil) and (copen == "<!--") and (cstart == "Start file")) then -- split the file
2661 io.output(newfilename) ;
2662 else -- not a splitpoint
```

```
2663 io.write (line .. "\n") ;
2664 end
2665 end -- do
2666 io.close(sfile)
2667 end -- function
2668
2669 -- Incorrect value, so print an error and exit.
2670
2671 function cvalueerror ( line, linenum , cvalue )
2672     print ( linenum .. " : " .. line ) ;
2673     print ("lwarpmk: incorrect variable value \" .. cvalue .. "\" in lwarpmk.conf.\n" ) ;
2674     printconf ( ) ;
2675     os.exit(1) ;
2676 end
2677
2678 -- Load settings from the project's "lwarpmk.conf" file:
2679
2680 function loadconf ( )
2681 -- Default configuration filename:
2682 local conffile = "lwarpmk.conf"
2683 -- Optional configuration filename:
2684 if arg[2] ~= nil then conffile = arg[2].."lwarpmkconf" end
2685 -- Default language:
2686 language = "english"
2687 -- Default xdyfile:
2688 xdyfile = "lwarp.xdy"
2689 -- Verify the file exists:
2690 if (lfs.attributes(conffile,"mode")==nil) then -- file not exists
2691 print("lwarpmk: " .. conffile .. " does not exist.")
2692 print("lwarpmk: " .. arg[2] .. " does not appear to be a project name.\n")
2693 printhelp ( ) ;
2694 os.exit(1) -- exit the entire lwarpmk script
2695 else -- file exists
2696 -- Read the file:
2697 print ("lwarpmk: Reading " .. conffile .. ".")
2698 local cfile = io.open(conffile)
2699 -- Scan each line:
2700 local linenum = 0
2701 for line in cfile:lines() do -- scan lines
2702 linenum = linenum + 1
2703 i,j,cvarname,cvalue = string.find (line,"([%w-]*)%s*=%s*"([%w%-_%.]*)\"") ;
2704 -- Error if incorrect enclosing characters:
2705 if ( i == nil ) then
2706 print ( linenum .. " : " .. line ) ;
2707 print ( "lwarpmk: Incorrect entry in " .. conffile .. ".\n" ) ;
2708 printconf ( ) ;
2709 os.exit(1) ;
2710 end
2711 if ( cvarname == "opssystem" ) then
2712     -- Verify choice of opssystem:
```

```
2713     if ( (cvalue == "Unix") or (cvalue == "Windows") ) then
2714         opsystem = cvalue
2715     else
2716         cvalueerror ( line, linenum , cvalue )
2717     end
2718 elseif ( cvarname == "latexname" ) then
2719     -- Verify choice of LaTeX compiler:
2720     if (
2721         (cvalue == "pdflatex") or
2722         (cvalue == "xelatex") or
2723         (cvalue == "lualatex")
2724     ) then
2725         latexname = cvalue
2726     else
2727         cvalueerror ( line, linenum , cvalue )
2728     end
2729 elseif ( cvarname == "sourcename" ) then sourcename = cvalue
2730 elseif ( cvarname == "homehtmlfilename" ) then homehtmlfilename = cvalue
2731 elseif ( cvarname == "htmlfilename" ) then htmlfilename = cvalue
2732 elseif ( cvarname == "latexmk" ) then latexmk = cvalue
2733 elseif ( cvarname == "language" ) then language = cvalue
2734 elseif ( cvarname == "xdyfile" ) then xdyfile = cvalue
2735 else
2736 print ( linenum .. " : " .. line ) ;
2737 print ("lwarpmk: Incorrect variable name \" .. cvarname .. "\" in \" .. conffile ..\".\n" ) ;
2738 printconf ( ) ;
2739 os.exit(1) ;
2740 end
2741 end -- do scan lines
2742 io.close(cfile)
2743 end -- file exists
2744 -- Select some operating-system commands:
2745 if opsystem=="Unix" then -- For Unix / Linux / Mac OS:
2746 rmname = "rm"
2747 mvname = "mv"
2748 touchnamepre = "touch"
2749 touchnamepost = ""
2750 dirslash = "/"
2751 opquote= "\"'"
2752 elseif opsystem=="Windows" then -- For Windows
2753 rmname = "DEL"
2754 mvname = "MOVE"
2755 touchnamepre = "COPY /b"
2756 touchnamepost = "+,,"
2757 dirslash = "\\\"
2758 opquote= "\"'"
2759 else print ( "lwarpmk: Select Unix or Windows for opsystem" )
2760 end --- for Windows
2761
2762 -- set xindycmd according to pdflatex vs xelatex/lualatex:
```

```
2763 if ( latexname == "pdflatex" ) then
2764 xindycmd = "texindy -C utf8"
2765 glossarycmd = "xindy -C utf8"
2766 else
2767 xindycmd = "xindy -M texindy -C utf8"
2768 glossarycmd = "xindy -C utf8"
2769 end
2770
2771 end -- loadconf
2772
2773
2774 function refreshdate ()
2775 os.execute(touchnamepre .. " " .. sourcename .. ".tex " .. touchnamepost)
2776 end
2777
2778
2779 -- Scan the LaTeX log file for the phrase "Rerun to get",
2780 -- indicating that the file should be compiled again.
2781 -- Return true if found.
2782
2783 function reruntoget (filesorce)
2784 local fsource = io.open(filesorce)
2785 for line in fsource:lines() do
2786 if ( string.find(line,"Rerun to get") ~= nil ) then
2787     io.close(fsource)
2788     return true
2789 end
2790 end
2791 io.close(fsource)
2792 return false
2793 end
2794
2795
2796 -- Compile one time, return true if should compile again.
2797 -- fsuffix is "" for print, "_html" for HTML output.
2798
2799 function onetime (fsuffix)
2800 print("lwarpmk: Compiling with " .. latexname .. " " .. sourcename..fsuffix)
2801 err = os.execute(
2802 --     "echo " ..
2803     latexname .. " " .. sourcename..fsuffix )
2804 if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2805 return (reruntoget(sourcename .. fsuffix .. ".log") ) ;
2806 end
2807
2808
2809 -- Compile up to five times.
2810 -- fsuffix is "" for print, "_html" for HTML output
2811
2812 function manytimes (fsuffix)
```

```
2813 if onetime(fsuffix) == true then
2814 if onetime(fsuffix) == true then
2815 if onetime(fsuffix) == true then
2816 if onetime(fsuffix) == true then
2817 if onetime(fsuffix) == true then
2818 end end end end end
2819 end
2820
2821 -- Exit if the given file does not exist.
2822
2823 function verifyfileexists (filename)
2824 if (lfs.attributes ( filename , "modification" ) == nil ) then
2825 print ( "lwarpmk: " .. filename .. " not found." );
2826 os.exit (1) ;
2827 end
2828 end
2829
2830
2831 -- Convert <project>_html.pdf into HTML files:
2832
2833 function pdftohtml ()
2834     -- Convert to text:
2835     print ("lwarpmk: Converting " .. sourcename
2836           .. "_html.pdf to " .. sourcename .. "_html.html")
2837     os.execute("pdftotext -enc UTF-8 -nopgbrk -layout "
2838               .. sourcename .. "_html.pdf " .. sourcename .. "_html.html")
2839     -- Split the result into individual HTML files:
2840     splitfile (homehtmlfilename .. ".html" , sourcename .. "_html.html")
2841 end
2842
2843
2844 -- Remove auxiliary files:
2845
2846 function removeaux ()
2847     os.execute ( rmname .. " " ..
2848                 sourcename .. ".aux " .. sourcename .. "_html.aux " ..
2849                 sourcename .. ".toc " .. sourcename .. "_html.toc " ..
2850                 sourcename .. ".lof " .. sourcename .. "_html.lof " ..
2851                 sourcename .. ".lot " .. sourcename .. "_html.lot " ..
2852                 sourcename .. ".idx " .. sourcename .. "_html.idx " ..
2853                 sourcename .. ".ind " .. sourcename .. "_html.ind " ..
2854                 sourcename .. ".log " .. sourcename .. "_html.log " ..
2855                 sourcename .. ".gl*" .. sourcename .. "_html.gl*" ..
2856                 "*_html_inc.*"
2857     )
2858 end
2859
2860
2861
2862 -- Create lateximages based on lateximages.txt:
```

```
2863 function createlateximages ()
2864 print ("lwarpmk: Creating lateximages.")
2865 local limagesfile = io.open("lateximages.txt")
2866 -- Create the lateximages directory, ignore error if already exists
2867 err = os.execute("mkdir lateximages")
2868 -- Scan lateximages.txt
2869 for line in limagesfile:lines() do
2870 -- lwimgpage is the page number in the PDF which has the image
2871 -- lwimgnum is the sequential lateximage number to assign for the image
2872 i,j,lwimgpage,lwimgnum = string.find (line,"|(.)|(.)|")
2873 -- For each entry:
2874 if ( i~=nil ) then
2875 -- Separate out the image into its own single-page pdf:
2876 err = os.execute(
2877 "pdfseparate -f " .. lwimgpage .. " -l " ..
2878 lwimgpage .. " " .. sourcename .. "_html.pdf lateximagetemp-%d.pdf")
2879 -- Crop the image:
2880 err = os.execute(
2881 "pdfcrop --hires lateximagetemp-" .. lwimgpage .. ".pdf lateximage-" .. lwimgnum .. ".pdf")
2882 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2883 -- Convert the image to svg:
2884 err = os.execute(
2885 "pdftocairo -svg lateximage-" .. lwimgnum .. ".pdf lateximage-" .. lwimgnum .. ".svg")
2886 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2887 -- Move the result into lateximages/:
2888 err = os.execute(
2889 mvname .. " lateximage-" .. lwimgnum .. ".svg lateximages" .. dirslash )
2890 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2891 -- Remove the temporary files:
2892 err = os.execute(
2893 rmname .. " lateximage-" .. lwimgnum .. ".pdf lateximagetemp-" .. lwimgpage .. ".pdf")
2894 if ( err ~= 0 ) then print ( "lwarpmk: File error." ) ; os.exit(1) ; end
2895 end
2896 end -- do
2897 io.close(limagesfile)
2898 end -- function
2899
2900
2901 -- Use latexmk to compile source and index:
2902 -- fsuffix is "" for print, or "_html" for HTML
2903 function compilelatexmk ( fsuffix )
2904 -- The recorder option is required to detect changes in <project>.tex
2905 -- while we are loading <project>_html.tex.
2906 err=os.execute ( "latexmk -pdf -dvi- -ps- -recorder "
2907 .. "-e "
2908 .. opquote
2909 .. "$makeindex = q/" -- $
2910 .. xindycmd
2911 .. " -M " .. xdyfile
2912 .. " -L " .. language .. " /"
```

```
2913     .. opquote
2914     .. " -pdflatex=\\" .. latexname .. " %O %S\" "
2915     .. sourcename..fsuffix ..".tex" );
2916     if ( err ~= 0 ) then print ( "lwarpmk: Compile error." ) ; os.exit(1) ; end
2917 end
2918
2919
2920
2921 -- lwarpmk --version :
2922
2923 if ( arg[1] == "--version" ) then
2924   print ( "lwarpmk: " .. printversion )
2925
2926 else -- not --version
2927
2928 -- print intro:
2929
2930 print ( "lwarpmk: " .. printversion .. " Automated make for the LaTeX lwarp package." )
2931
2932 -- lwarpmk print:
2933
2934 if arg[1] == "print" then
2935   loadconf ( )
2936   if ( latexmk == "true" ) then
2937     compilelatexmk ( "" )
2938     print ( "lwarpmk: Done." )
2939   else -- not latexmk
2940     verifyfileexists ( sourcename .. ".tex" ) ;
2941     -- See if up to date:
2942     if (
2943       ( lfs.attributes ( sourcename .. ".pdf" , "modification" ) == nil ) or
2944       (
2945         lfs.attributes ( sourcename .. ".tex" , "modification" ) >
2946         lfs.attributes ( sourcename .. ".pdf" , "modification" )
2947       )
2948     ) then
2949       -- Recompile if not yet up to date:
2950       manytimes( "" )
2951       print ( "lwarpmk: Done." ) ;
2952     else
2953       print ( "lwarpmk: " .. sourcename .. ".pdf is up to date." ) ;
2954     end
2955   end -- not latexmk
2956
2957 elseif arg[1] == "print1" then
2958   loadconf ( )
2959   verifyfileexists ( sourcename .. ".tex" ) ;
2960   onetime( "" )
2961   print ( "lwarpmk: Done." ) ;
2962
```

```
2963 -- lwarp printindex:
2964 -- Compile the index then touch the source
2965 -- to trigger a recompile of the document:
2966
2967 elseif arg[1] == "printindex" then
2968 loadconf ()
2969 print ("lwarpmk: Processing the index.")
2970 os.execute(
2971     xindycmd
2972     .. " -M " .. xdyfile
2973     .. " -L " .. language
2974     .. " " .. sourcename .. ".idx")
2975 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2976 refreshdate ()
2977 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2978 print ("lwarpmk: Done.")
2979
2980 -- lwarp printglossary:
2981 -- Compile the glossary then touch the source
2982 -- to trigger a recompile of the document:
2983
2984 elseif arg[1] == "printglossary" then
2985 loadconf ()
2986 print ("lwarpmk: Processing the glossary.")
2987
2988 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
2989     " -t " .. sourcename .. ".glg -o " .. sourcename .. ".gls "
2990     .. sourcename .. ".glo")
2991 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
2992 refreshdate ()
2993 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
2994 print ("lwarpmk: Done.")
2995
2996 -- lwarpmk html:
2997
2998 elseif arg[1] == "html" then
2999 loadconf ()
3000 if ( latexmk == "true" ) then
3001     compilelatexmk ("_html")
3002     pdftohtml ()
3003     print ("lwarpmk: Done.")
3004 else -- not latexmk
3005     verifyfileexists ( sourcename .. ".tex" ) ;
3006     -- See if exists and is up to date:
3007     if (
3008         ( lfs.attributes ( homehtmlfilename .. ".html" , "modification" ) == nil ) or
3009         (
3010             lfs.attributes ( sourcename .. ".tex" , "modification" ) >
3011             lfs.attributes ( homehtmlfilename .. ".html" , "modification" )
3012         )
3013     )
```

```
3013 ) then
3014     -- Recompile if not yet up to date:
3015     manytimes("_html")
3016     pdftohtml ()
3017     print ("lwarpmk: Done.")
3018 else
3019     print ("lwarpmk: " .. homehtmlfilename .. ".html is up to date.")
3020 end
3021 end -- not latexmk
3022
3023 elseif arg[1] == "html1" then
3024     loadconf ()
3025     verifyfileexists ( sourcename .. ".tex" );
3026     onetime("_html")
3027     pdftohtml ()
3028     print ("lwarpmk: Done.")
3029
3030 elseif arg[1] == "pdftohtml" then
3031     loadconf ()
3032     pdftohtml ()
3033
3034 -- lwarpmk htmlindex:
3035 -- Compile the index then touch the source
3036 -- to trigger a recompile of the document:
3037
3038 elseif arg[1] == "htmlindex" then
3039 loadconf ()
3040 print ("lwarpmk: Processing the index.")
3041 os.execute(
3042     xindycmd
3043     .. " -M " .. xdyfile
3044     .. " -L " .. language
3045     .. " " .. sourcename .. "_html.idx"
3046 )
3047 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3048 refreshdate ()
3049 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3050 print ("lwarpmk: Done.")
3051
3052 -- lwarpmk htmlglossary:
3053 -- Compile the glossary then touch the source
3054 -- to trigger a recompile of the document:
3055
3056 elseif arg[1] == "htmlglossary" then
3057 loadconf ()
3058 print ("lwarpmk: Processing the glossary.")
3059
3060 os.execute(glossarycmd .. " -L " .. language .. " -I xindy -M " .. sourcename ..
3061     "_html -t " .. sourcename .. "_html.glg -o " .. sourcename ..
3062     "_html.gls " .. sourcename .. "_html.glo")
```

```
3063
3064 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3065 refreshdate ()
3066 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3067 print ("lwarpmk: Done.")
3068
3069 -- lwarpmk limages:
3070 -- Scan the lateximages.txt file to create lateximages,
3071 -- then touch the source to trigger a recompile.
3072
3073 elseif arg[1] == "limages" then
3074 loadconf ()
3075 print ("lwarpmk: Processing images.")
3076 createlateximages ()
3077 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3078 refreshdate ()
3079 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3080 print ("lwarpmk: Done.")
3081
3082 -- lwarpmk again:
3083 -- Touch the source to trigger a recompile.
3084
3085 elseif arg[1] == "again" then
3086 loadconf ()
3087 print ("lwarpmk: Forcing an update of " .. sourcename .. ".tex.")
3088 refreshdate ()
3089 print ("lwarpmk: " .. sourcename .. ".tex is ready to be recompiled.")
3090 print ("lwarpmk: Done.")
3091
3092 -- lwarpmk clean:
3093 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3094
3095 elseif arg[1] == "clean" then
3096 loadconf ()
3097 removeaux ()
3098 print ("lwarpmk: Done.")
3099
3100 -- lwarpmk cleanall
3101 -- Remove project.aux, .toc, .lof, .lot, .idx, .ind, .log, *_html_inc.*, .gl*
3102 -- and also project.pdf, *.html
3103
3104 elseif arg[1] == "cleanall" then
3105 loadconf ()
3106 removeaux ()
3107 os.execute ( rmname .. " " ..
3108     sourcename .. ".pdf " .. sourcename .. "_html.pdf " ..
3109     "*.html"
3110 )
3111 print ("lwarpmk: Done.")
3112
```

```

3113 -- lwarpmk with no argument :
3114
3115 elseif (arg[1] == nil) then
3116 printhelp ()
3117
3118 -- lwarpmk -h or lwarpmk --help :
3119
3120 elseif (arg[1] == "-h" ) or (arg[1] == "--help") then
3121 printusage ()
3122
3123 else
3124 print ("lwarpmk: Unknown command \""..arg[1].."\".\n")
3125 printhelp ()
3126 end
3127
3128 end -- not --version
3129 \end{filecontents*}
3130 % \end{Verbatim}% for syntax highlighting

3131 \end{LWR@createlwarpmk}

```

31 Stacks

for HTML output: 3132 \begin{warpHTML}



Stacks are used to remember how to close sections and list items. Before a new section is started, previously nested sections and items must be closed out (un-nested) in proper order. Note that starting a new section may close several levels of previously nested items at the same time. For example, starting a new `\section` would close any currently open subsection, subsubsection, and paragraph. General environments are not nested on the stack since they have their own close mechanism. List environments are nested, and items inside those environments are nested one level deeper still. List environments may be nested inside other list environments, and list items are nested inside list environments as well. Thus, the stack may have items which are not necessarily in order, since a description may contain an enumerate, for example. Depths to be recorded in `\LWR@closedepthone`, etc.

31.1 Assigning depths

initial depths for empty stack entries:

```
3133 \newcommand*{\LWR@depthnone}{-5}
```

all sectioning depths are deeper than LWR@depthfinished:

```
3134 \newcommand*\LWR@depthfinished}{-4}
3135 \newcommand*\LWR@depthpart}{-1}
3136 \newcommand*\LWR@depthchapter}{0}
3137 \newcommand*\LWR@depthsection}{1}
3138 \newcommand*\LWR@depthsubsection}{2}
3139 \newcommand*\LWR@depthsubsubsection}{3}
3140 \newcommand*\LWR@depthparagraph}{4}
3141 \newcommand*\LWR@depthsubparagraph}{5}
```

used by \itemize, \enumerate, \description:

```
3142 \newcommand*\LWR@depthlist}{6}
```

used by \item:

```
3143 \newcommand*\LWR@depthlistitem}{7}
```

31.2 Closing actions

A stack to record the action to take to close each nesting level: Add more levels of stack if necessary for a very deeply nested document, adding to \pushclose and \popclose as well.

```
3144 \newcommand*\LWR@closeone}{}% top of the stack
3145 \newcommand*\LWR@closetwo}{ }
3146 \newcommand*\LWR@closethree}{ }
3147 \newcommand*\LWR@closefour}{ }
3148 \newcommand*\LWR@closefive}{ }
3149 \newcommand*\LWR@closesix}{ }
3150 \newcommand*\LWR@closeseven}{ }
3151 \newcommand*\LWR@closeeight}{ }
3152 \newcommand*\LWR@closenine}{ }
3153 \newcommand*\LWR@closeten}{ }
3154 \newcommand*\LWR@closeeleven}{ }
3155 \newcommand*\LWR@closetwelve}{ }
```

31.3 Closing depths

A stack to record the depth of each level:



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

Ex:

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

```
3156 \newcommand*{\LWR@closedepthone}{\LWR@depthnone}% top of the stack
3157 \newcommand*{\LWR@closedepthtwo}{\LWR@depthnone}
3158 \newcommand*{\LWR@closedepththree}{\LWR@depthnone}
3159 \newcommand*{\LWR@closedepthfour}{\LWR@depthnone}
3160 \newcommand*{\LWR@closedepthfive}{\LWR@depthnone}
3161 \newcommand*{\LWR@closedepthsix}{\LWR@depthnone}
3162 \newcommand*{\LWR@closedepthseven}{\LWR@depthnone}
3163 \newcommand*{\LWR@closedeptheight}{\LWR@depthnone}
3164 \newcommand*{\LWR@closedepthnine}{\LWR@depthnone}
3165 \newcommand*{\LWR@closedephten}{\LWR@depthnone}
3166 \newcommand*{\LWR@closedeptheleven}{\LWR@depthnone}
3167 \newcommand*{\LWR@closedephtwelve}{\LWR@depthnone}
```

31.4 Pushing and popping the stack

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

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

```
3168 \NewDocumentCommand{\pushclose}{m m}
3169 {
3170 \let\LWR@closetwelve\LWR@closeeleven
3171 \let\LWR@closeeleven\LWR@closeten
3172 \let\LWR@closeten\LWR@closenine
3173 \let\LWR@closenine\LWR@closeeight
3174 \let\LWR@closeeight\LWR@closeseven
3175 \let\LWR@closeseven\LWR@closesix
3176 \let\LWR@closesix\LWR@closefive
3177 \let\LWR@closefive\LWR@closefour
3178 \let\LWR@closefour\LWR@closethree
3179 \let\LWR@closethree\LWR@closetwo
3180 \let\LWR@closetwo\LWR@closeone
3181 \let\LWR@closeone#1
3182 \let\LWR@closedephtwelve\LWR@closedeptheleven
3183 \let\LWR@closedeptheleven\LWR@closedephten
3184 \let\LWR@closedephten\LWR@closedepthnine
```

```

3185 \let\LWR@closedepthnine\LWR@closedeptheight
3186 \let\LWR@closedeptheight\LWR@closedeptheven
3187 \let\LWR@closedeptheven\LWR@closedepthsix
3188 \let\LWR@closedepthsix\LWR@closedepthfive
3189 \let\LWR@closedepthfive\LWR@closedepthfour
3190 \let\LWR@closedepthfour\LWR@closedepththree
3191 \let\LWR@closedepththree\LWR@closedepthtwo
3192 \let\LWR@closedepthtwo\LWR@closedepthone
3193 \let\LWR@closedepthone#2
3194 }

```

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

```

3195 \newcommand*{\popclose}
3196 {
3197 \let\LWR@closeone\LWR@closetwo
3198 \let\LWR@closetwo\LWR@closethree
3199 \let\LWR@closethree\LWR@closefour
3200 \let\LWR@closefour\LWR@closefive
3201 \let\LWR@closefive\LWR@closesix
3202 \let\LWR@closesix\LWR@closeseven
3203 \let\LWR@closeseven\LWR@closeeight
3204 \let\LWR@closeeight\LWR@close-nine
3205 \let\LWR@close-nine\LWR@close-ten
3206 \let\LWR@close-ten\LWR@close-eleven
3207 \let\LWR@close-eleven\LWR@close-twelve
3208 \let\LWR@closedepthone\LWR@closedepthtwo
3209 \let\LWR@closedepthtwo\LWR@closedepththree
3210 \let\LWR@closedepththree\LWR@closedepthfour
3211 \let\LWR@closedepthfour\LWR@closedepthfive
3212 \let\LWR@closedepthfive\LWR@closedepthsix
3213 \let\LWR@closedepthsix\LWR@closedepthseven
3214 \let\LWR@closedepthseven\LWR@closedeptheight
3215 \let\LWR@closedeptheight\LWR@closedepthnine
3216 \let\LWR@closedepthnine\LWR@closedephten
3217 \let\LWR@closedephten\LWR@closedeptheleven
3218 \let\LWR@closedeptheleven\LWR@closedephtwelve
3219 }

3220 \end{warpHTML}

```

32 Data arrays

These macros are similar to the `arrayjobx` package, except that `\LWR@setexpararray'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: 3221 `\begin{warpHTML}`

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

```
3222 \NewDocumentCommand{\LWR@setexparray}{m m m}{%
3223 \ifthenelse{\isempty{#3}}{%
3224 {\csdef{#1#2}{}}
3225 {\expandafter\edef\csname #1#2\endcsname{\expandonce#3}}%
3226 }
```

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

```
3227 \newcommand*\LWR@getexparray}[2]{\csuse{#1#2}}
```

```
3228 \end{warpHTML}
```

33 HTML entities

for HTML output: 3229 `\begin{warpHTML}`

HTML entites and HTML Unicode entities:

```
3230 \let\LWR@origampersand\&
```

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

```
3231 \newcommand*\HTMLentity}[1]{%
3232 % \LWR@traceinfo{HTMLentity \detokenize{#1}}%
3233 \begingroup%
3234 \LWR@FBcancel%
3235 \LWR@origampersand#1;%
3236 \endgroup
3237 % \LWR@traceinfo{HTMLentity done}%
3238 }
```

```
\HTMLunicode {<hex_unicode>}
```

```
3239 \newcommand*\HTMLunicode}[1]{\HTMLentity{\#x#1}}
```

\&

```
3240 \renewcommand*{\&}{\HTMLentity{amp}}
```

\textless

\textgreater

```
3241 \let\LWR@origtextless\textless
3242 \renewcommand*{\textless}{\HTMLentity{lt}}
3243
3244 \let\LWR@origtextgreater\textgreater
3245 \renewcommand*{\textgreater}{\HTMLentity{gt}}
```

```
3246 \end{warpHTML}
```

34 HTML filename generation

The filename of the homepage is set to `\HomeHTMLFilename.html`. The filenames of additional sections start with `\HTMLFilename`, to which is appended a section number or a simplified section name, depending on `FileSectionNames`.

for HTML & PRINT: 3247 `\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.

```
3248 \providecommand*{\BaseJobname}{\jobname}
```

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

```
3249 \providecommand*{\HTMLFilename}{}
```

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

```
3250 \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.

```
3251 \newcommand*\SetHTMLFileNumber}[1]{%
3252 \setcounter{LWR@htmlfilenumber}{#1}%
3253 }
```

Bool FileSectionNames Selects how to create HTML file names.

Defaults to use section names in the filenames.

```
3254 \newbool{FileSectionNames}
3255 \booltrue{FileSectionNames}
```

```
3256 \end{warpall}
```

for HTML output: 3257 \begin{warpHTML}

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

```
3258 \newcounter{LWR@htmlfilenumber}
3259 \setcounter{LWR@htmlfilenumber}{0}
```

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

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

```
3260 \newcommand*\LWR@htmlsectionfilename}[1]{%
3261 \LWR@traceinfo{LWR@htmlsectionfilename A}%
```

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

```
3262 \LWR@traceinfo{about to assign temp}%
3263 \edef\LWR@tempone{#1}%
3264 \LWR@traceinfo{about to compare with ??}%
3265 \ifthenelse{\equal{\LWR@tempone}{??}}{%
3266 {\LWR@traceinfo{found ??}}%
3267 {\LWR@traceinfo{not found ??}}%
3268 \LWR@traceinfo{about to compare with zero or empty}%
3269 \ifthenelse{%
3270   \equal{\LWR@tempone}{0}%
3271   \OR \equal{\LWR@tempone}{}%
3272   \OR \equal{\LWR@tempone}{??}}%
3273 }%
3274 {%
```

```

3275 \LWR@traceinfo{LWR@htmlsectionfilename B \HomeHTMLFilename.html}%
3276 \HomeHTMLFilename.html%
3277 }%

```

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

```

3278 {%
3279 \LWR@traceinfo{LWR@htmlsectionfilename C \LWR@tempone}%
3280 \ifthenelse{%
3281 \equal{\HTMLFilename}{ } \AND
3282 \equal{\LWR@tempone}{Index} \OR
3283 \equal{\LWR@tempone}{index}}%
3284 }%
3285 {%
3286 \LWR@traceinfo{prefixing the index name with an underscore.}%
3287 \_#1.html%
3288 }%

```

Otherwise, create a filename with the chosen prefix:

```

3289 {\HTMLFilename#1.html}%
3290 }%
3291 \LWR@traceinfo{LWR@htmlsectionfilename Z}%
3292 }

```

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

Prints the filename for the given label

```

3293 \newcommand*{\LWR@htmlrefsectionfilename}[1]{%
3294 \LWR@traceinfo{LWR@htmlrefsectionfilename A: !#1!}%
3295 \LWR@htmlsectionfilename{\LWR@htmlfileref{#1}}%
3296 \LWR@traceinfo{LWR@htmlrefsectionfilename B}%
3297 }

3298 \end{warpHTML}

```

35 Homepage link

for HTML output: 3299 `\begin{warpHTML}`

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

```

3300 \newcommand*{\LinkHome}{%
3301 \LWR@subhyperrefclass{%
3302 \HomeHTMLFilename.html}%
3303 {Home}{linkhome}%
3304 }

```

`\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.

```

3305 \newcommand*{\LWR@topnavigation}{
3306 \LWR@htmlclassline{nav}{topnavigation}{\LinkHome}
3307 }

```

`\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.

```

3308 \newcommand*{\LWR@botnavigation}{
3309 \LWR@htmlclassline{nav}{botnavigation}{\LinkHome}
3310 }

```

```

3311 \end{warpHTML}

```

36 \PrintStack diagnostic tool



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

for HTML output: 3312 `\begin{warpHTML}`

`\PrintStack` Prints the closedepth stack.

```

3313 \newcommand*{\PrintStack}{
3314 \LWR@startpars
3315 \LWR@closedepthone{} \LWR@closedepthtwo{} \LWR@closedepththree{}
3316 \LWR@closedepthfour{} \LWR@closedepthfive{} \LWR@closedepthsix{}
3317 \LWR@closedepthseven{} \LWR@closedeptheight{} \LWR@closedepthnine{}
3318 \LWR@closedephten{} \LWR@closedeptheleven{} \LWR@closedephtwelve{}
3319 }

```

```

3320 \end{warpHTML}

```

37 Closing stack levels

for HTML output: 3321 `\begin{warpHTML}`

Close one nested level:

```
3322 \newcommand*{\LWR@closeoneprevious}{%
3323
3324 \LWR@closeone{}
3325
3326 \popclose{}
3327 }
```

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

```
3328 \newcommand*{\LWR@closeprevious}[1]{
```

Close any pending paragraph:

```
3329 \LWR@stoppars
```

Close anything nested deeper than the desired depth:

```
3330 \whiledo{\not\(\LWR@closedepthone<#1\)}{\LWR@closeoneprevious}
3331 }
```

```
3332 \end{warpHTML}
```

38 PDF pages and styles

for HTML output: 3333 `\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.

```
3334 \newcommand{\LWR@forcenewpage}{%
3335 \LWR@stoppars\LWR@orignewpage\LWR@startpars%
3336 }
```

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

```
\pagestyle   {<style>}
              3337 \renewcommand*{\pagestyle}[1]{}

\thispagestyle {<style>}
              3338 \renewcommand*{\thispagestyle}[1]{}

\markboth   {<left>} {<right>}
              3339 \renewcommand*{\markboth}[2]{}

\markright  {<right>}
              3340 \renewcommand*{\markright}[1]{}

\raggedbottom
              3341 \renewcommand*{\raggedbottom}{}

\flushbottom
              3342 \renewcommand*{\flushbottom}{}

\sloppy
              3343 \renewcommand*{\sloppy}{}

\fussy
              3344 \renewcommand*{\fussy}{}

\pagenumbering {<commands>}
              3345 \renewcommand*{\pagenumbering}[1]{}
              3346 \end{warpHTML}
```

39 HTML tags, spans, divs, elements

for HTML output: 3347 \begin{warpHTML}

39.1 Mapping L^AT_EX Sections to HTML Sections

```

3348 \newcommand*\LWR@tagpart}{h2}
3349 \newcommand*\LWR@tagpartend}{/h2}
3350 \newcommand*\LWR@tagchapter}{h3}
3351 \newcommand*\LWR@tagchapterend}{/h3}
3352 \newcommand*\LWR@tagsection}{h4}
3353 \newcommand*\LWR@tagsectionend}{/h4}
3354 \newcommand*\LWR@tagsubsection}{h5}
3355 \newcommand*\LWR@tagsubsectionend}{/h5}
3356 \newcommand*\LWR@tagsubsubsection}{h6}
3357 \newcommand*\LWR@tagsubsubsectionend}{/h6}
3358 \newcommand*\LWR@tagparagraph}{span class="paragraph"{} }
3359 \newcommand*\LWR@tagparagraphend}{/span}
3360 \newcommand*\LWR@tagsubparagraph}{span class="subparagraph"{} }
3361 \newcommand*\LWR@tagsubparagraphend}{/span}
3362
3363 \newcommand*\LWR@tagregularparagraph}{p}

```

39.2 Babel-French

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

(Based on original code by DANIEL FLIPO.)

```

3364 \providecommand*\LWR@FBcancel{}
3365
3366 \AtBeginDocument{
3367 \@ifundefined{frenchbsetup}
3368 {}
3369 {
3370   \frenchbsetup{FrenchFootnotes=false}
3371   % OR: redefine \insertfootnotemarkFB?
3372   \LetLtxMacro\LWR@FBcancel\NoAutoSpacing
3373   \renewcommand*\FBcolonspace}{%
3374     \begingroup%
3375     \LWR@FBcancel%
3376     \LWR@origampersand{}nosp;%
3377     \endgroup%
3378   }
3379   \renewcommand*\FBthinspace}{%
3380     \begingroup%
3381     \LWR@FBcancel%
3382     \LWR@origampersand\#x202f;% \,
3383     \endgroup%
3384   }
3385   \renewcommand*\FBguillspace}{%

```

```

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

```

39.3 HTML tags

`\LWR@htmltagc` `{<tag>}` Break ligatures and use upright apostrophes in HTML tags.

`\protect` is in case the tag appears in TOC, LOF, LOT.

```

3411 \newcommand*{\LWR@htmltagc}[1]{%
3412 {%
3413 % \LWR@traceinfo{\LWR@htmltagc \detokenize{#1}}%
3414 \begingroup%
3415 \LWR@FBcancel%
3416 \ifmmode\else\protect\LWR@origttfamily\fi%
3417 \protect\LWR@origtextless#1\protect\LWR@origtextgreater%
3418 \endgroup%
3419 }%
3420 }

```

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.

```

3421 \newcommand*{\LWR@nestspanitem}{%
3422 \ifnewlist\else{\LWR@htmltagc{br /}}\fi%
3423 \LWR@origitem%
3424 }
3425
3426 \newenvironment*{LWR@nestspan}
3427 {%
3428 \LWR@traceinfo{LWR@nestspan}%
3429 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3430 {%
3431   \LWR@traceinfo{LWR@nestspan: inside a lateximage}%
3432 }%
3433 {% not in a lateximage
3434   \LWR@traceinfo{LWR@nestspan: NOT inside a lateximage}%
3435   \addtocounter{LWR@spandepth}{1}%
3436   \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
3437   \RenewDocumentEnvironment{BlockClass}{o m}{-}{-}%
3438   \renewcommand{\BlockClassSingle}[2]{##2}%
3439   \renewcommand{\LWR@forcenewpage}{-}%
3440   \renewcommand{\LWR@itemizestart}{-}%
3441   \let\item\LWR@nestspanitem%
3442   }%
3443   \renewcommand{\LWR@itemizeend}{-}%
3444   \renewcommand{\LWR@enumeratestart}{-}%
3445   \let\item\LWR@nestspanitem%
3446   }%
3447   \renewcommand{\LWR@enumerateend}{-}%
3448   \renewcommand{\LWR@descriptionstart}{-}%
3449   \let\item\LWR@nestspanitem%
3450   }%
3451   \renewcommand{\LWR@descriptionend}{-}%
3452 }% not in a lateximage
3453 }% starting env
3454 }% ending env
3455 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3456 {}%
3457 {\addtocounter{LWR@spandepth}{-1}}%
3458 \LWR@traceinfo{LWR@nestspan: done}%
3459 }
3460
3461 \AfterEndEnvironment{LWR@nestspan}{\global\let\par\LWR@closeparagraph}

```

`\LWR@htmlspan` `{<tag>}{<text>}`

⚠ `\LWR@spandepth` is used to ensure that paragraph tags are not generated inside a span. The exact sequence of when to add and subtract the counter is important to

correctly handle the paragraph tags before and after the span.

```

3462 \NewDocumentCommand{\LWR@htmlspan}{m +m}{%
3463 \LWR@ensuredoingapar%
3464 \LWR@htmltagc{#1}%
3465 \begin{LWR@nestspan}%
3466 #2%
3467 \LWR@htmltagc{/#1}%
3468 \end{LWR@nestspan}%
3469 }

```

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

```

3470 \NewDocumentCommand{\LWR@htmlspanclass}{o m +m}{%
3471 \LWR@traceinfo{LWR@htmlspanclass #2}%
3472 \LWR@ensuredoingapar%
3473 \LWR@subhtmelementclass{span}[#1]{#2}%
3474 \begin{LWR@nestspan}%
3475 #3%
3476 \LWR@htmltagc{/span}%
3477 \LWR@traceinfo{LWR@htmlspanclass done}%
3478 \end{LWR@nestspan}%
3479 }

```

`\LWR@htmltag` {*tag*}

Print an HTML tag: <tag>

```

3480 \newcommand*{\LWR@htmltagb}[1]{%
3481 \LWR@htmltagc{#1}%
3482 \endgroup%
3483 }
3484
3485 \newcommand*{\LWR@htmltag}{%
3486 \begingroup\catcode'\_ =12
3487 \LWR@htmltagb%
3488 }

```

39.4 Block tags and comments

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

```

\LWR@htmlopencomment
\LWR@htmlclosecomment

```

```

3489 \newcommand*{\LWR@htmlopencomment}{%
3490 {%
3491 % \LWR@traceinfo{\LWR@htmlopencomment}%
3492 \begingroup%
3493 \LWR@FBcancel%
3494 \ifmmode\else\protect\LWR@origttfamily\fi%
3495 \LWR@origtextless{!}{-}{-}%
3496 \endgroup%
3497 }%
3498 }
3499
3500 \newcommand*{\LWR@htmlclosecomment}{%
3501 {%
3502 % \LWR@traceinfo{\LWR@htmlclosecomment}%
3503 \begingroup%
3504 \LWR@FBcancel%
3505 \ifmmode\else\protect\LWR@origttfamily\fi%
3506 {-}{-}\LWR@origtextgreater{}}%
3507 \endgroup%
3508 }%
3509 }

```

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

```

3510 \newcommand{\LWR@htmlcomment}[1]{%
3511 \LWR@htmlopencomment}%
3512 {%
3513 \LWR@origttfamily% break ligatures
3514 #1%
3515 }%
3516 \LWR@htmlclosecomment{}}

```

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

```

3517 \newcommand{\LWR@htmlblockcommentb}[1]
3518 {\LWR@stoppars\LWR@htmlcomment{#1}\LWR@startpars\endgroup}
3519
3520 \newcommand{\LWR@htmlblockcomment}
3521 {%
3522 \begingroup\catcode'\_ =12%
3523 \LWR@htmlblockcommentb%
3524 }

```

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

```

3525 \newcommand*{\LWR@htmlblocktag}[1]{%
3526 \LWR@stoppars%

```

```

3527 \LWR@htmltag{#1}%
3528 \LWR@startpars%
3529 }

```

39.5 Div class and element class

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

Factored and reused in several places.

The trailing spaces allow more places for a line break.

```

3530 \NewDocumentCommand{\LWR@subhtmlclass}{m o m}{%
3531 \LWR@traceinfo{\LWR@subhtmlclass #1 #3}%
3532 \IfValueTF{#2}%
3533 {% option
3534   \ifthenelse{\equal{#2}{}}%
3535   {\LWR@htmltag{#1 class="#3"}}% empty option
3536   {\LWR@htmltag{#1 class="#3" style="#2"}}% non-empty option
3537 }% option
3538 {\LWR@htmltag{#1 class="#3"}}% no option
3539 \LWR@traceinfo{\LWR@subhtmlclass done}%
3540 }

```

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

```

3541 \NewDocumentCommand{\LWR@htmlclass}{m o m}{%
3542 \LWR@stoppars%
3543 \LWR@subhtmlclass{#1}[#2]{#3}%
3544 \LWR@startpars%
3545 }

```

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

```

3546 \newcommand*{\LWR@htmlclassend}[2]{%
3547 \LWR@stoppars%
3548 \LWR@htmltag{/#1}%
3549 \ifbool{HTMLDebugComments}{%
3550   \LWR@htmlcomment{End of #1 ‘‘#2’’}%
3551 }{}%
3552 \LWR@startpars%
3553 }

```

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

```

3554 \NewDocumentCommand{\LWR@htmldivclass}{o m}{%
3555 \LWR@htmlclass{div}{#1}{#2}%
3556 }

```

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

```

3557 \newcommand*{\LWR@htmldivclassend}[1]{%
3558 \LWR@htmlclassend{div}{#1}%
3559 }

```

39.6 Single-line elements

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

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

```

3560 \NewDocumentCommand{\LWR@htmlclassline}{m o m +m}{%
3561 \LWR@stoppars
3562 \LWR@subhtmlclass{#1}{#2}{#3}%
3563 #4%
3564 \LWR@htmltag{/#1}
3565 \LWR@startpars
3566 }

```

39.7 HTML5 semantic elements

`\LWR@htmllement` $\{\langle element \rangle\}$

```

3567 \newcommand*{\LWR@htmllement}[1]{%
3568 \LWR@htmlblocktag{#1}
3569 }

```

`\LWR@htmllementend` $\{\langle element \rangle\}$

```

3570 \newcommand*{\LWR@htmllementend}[1]{%
3571 \LWR@stoppars
3572 \LWR@htmltag{/#1}
3573 \LWR@startpars
3574 }
3575
3576 \end{warpHTML}

```

39.8 High-level block and inline classes

These are high-level commands which allow the creation of arbitrary block or inline sections which may be formatted with css.

Nullified versions are provided for print mode.

For other direct-formatting commands, see section 72.

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

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

for HTML output: 3577 `\begin{warpHTML}`
 3578 `\NewDocumentEnvironment{BlockClass}{o m}{}`
 3579 `{`
 3580 `\LWR@htmldivclass[#1]{#2}`
 3581 `}`
 3582 `{`
 3583 `\LWR@htmldivclassend{#2}`
 3584 `}`
 3585 `\end{warpHTML}`

for PRINT output: 3586 `\begin{warpprint}`
 3587 `\NewDocumentEnvironment{BlockClass}{o m}{-}{-}`
 3588 `\end{warpprint}`

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

for HTML output: 3589 `\begin{warpHTML}`
 3590 `\newcommand{\BlockClassSingle}[2]{%`
 3591 `\LWR@htmlclassline{div}{#1}{#2}%`
 3592 `}`
 3593 `\end{warpHTML}`

for PRINT output: 3594 `\begin{warpprint}`
 3595 `\newcommand{\BlockClassSingle}[2]{#2}`
 3596 `\end{warpprint}`

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

for HTML output: 3597 `\begin{warpHTML}`
 3598 `\NewDocumentCommand{\InlineClass}{o m +m}{%`
 3599 `\LWR@htmlspanclass[#1]{#2}{#3}%`
 3600 `}`
 3601 `\end{warpHTML}`

for PRINT output: 3602 `\begin{warpprint}`

```
3603 \NewDocumentCommand{\InlineClass}{o m +m}{#3}%
3604 \end{warpprint}
```

39.9 Closing HTML tags

for HTML output: 3605 \begin{warpHTML}

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

```
3606 \newcommand*{\LWR@printclosepart}
3607   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing part}}{}}
3608 \newcommand*{\LWR@printclosechapter}
3609   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing chapter}}{}}
3610 \newcommand*{\LWR@printclosesection}
3611   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing section}}{}}
3612 \newcommand*{\LWR@printclosesubsection}
3613   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsection}}{}}
3614 \newcommand*{\LWR@printclosesubsubsection}
3615   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subsubsection}}{}}
3616 \newcommand*{\LWR@printcloseparagraph}
3617   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing paragraph}}{}}
3618 \newcommand*{\LWR@printclose subparagraph}
3619   {\ifbool{HTMLDebugComments}{\LWR@htmlcomment{Closing subparagraph}}{}}
```

Lists require closing HTML tags:

```
3620 \newcommand*{\LWR@printcloselistitem}
3621   {\LWR@htmltag{/li}}
3622 \newcommand*{\LWR@printclosedescitem}
3623   {\LWR@htmltag{/dd}}
3624 \newcommand*{\LWR@printcloseitemize}
3625   {\LWR@htmltag{/ul}}
3626 \newcommand*{\LWR@printcloseenumerate}
3627   {\LWR@htmltag{/ol}}
3628 \newcommand*{\LWR@printclosedescription}
3629   {\LWR@htmltag{/dl}}
```

```
3630 \end{warpHTML}
```

40 Paragraph handling

These commands generate the HTML paragraph tags when allowed and required.

Paragraph tags are or are not allowed depending on many conditions. Section 41 has high-level commands which allow paragraph-tag generation to start/stop. Even when allowed (`\LWR@doingstartpars`), tags are not generated until a \LaTeX paragraph is being used (`\LWR@doingapar`). `LWR@lateximagedepth` is used to prevent nesting tags inside a `lateximage`. `LWR@spandepth` is used to prevent nesting paragraph tags inside a paragraph, which became important inside `\fbox` commands and other spans.

for HTML output: 3631 `\begin{warpHTML}`

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

```
3632 \newcounter{LWR@spandepth}
3633 \setcounter{LWR@spandepth}{0}
```

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

```
3634 \newbool{LWR@doingstartpars}
3635 \boolfalse{LWR@doingstartpars}
```

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

```
3636 \newbool{LWR@doingapar}
3637 \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:

```
3638 \newcommand*{\LWR@ensuredoingapar}{%
3639 \ifbool{LWR@doingstartpars}%
3640 {\global\booltrue{LWR@doingapar}}%
3641 {}}%
3642 }
```

`\LWR@openparagraph`

```
3643 \newcommand*{\LWR@openparagraph}
3644 {%
```

See if paragraph handling is enabled:

```
3645 \ifbool{LWR@doingstartpars}%
3646 {% handling pars
```

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

```

3647 \iffthenelse{%
3648     \cntttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3649     \cntttest{\value{LWR@spandepth}}{>}{0}%
3650 }% 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 .

```

3651 {}% no nested par tags

```

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

```

3652 {% yes nest par tags
3653     \LWR@htmltagc{\LWR@tagregularparagraph}%

```

Manually indent item list labels to avoid left margin intrusion:

See if are nested inside an item list:

```

3654     \ifnumcomp{\@listdepth}{>}{0}%
3655     {%

```

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

```

3656     \LWR@origspace{1in}%
3657     }%
3658     {}%

```

Now have started a paragraph.

```

3659     \global\booltrue{LWR@doingapar}%

```

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

```

3660     \let\par\LWR@closeparagraph%
3661     }% end of yes nest par tags
3662 }% end of handling pars
3663 {}% not handling pars
3664 }

```

\LWR@closeparagraph

```

3665 \newcommand*{\LWR@closeparagraph}
3666 {%

```

See if paragraph handling is enabled:

```
3667 \ifbool{LWR@doingapar}%
```

If currently in paragraph mode:

```
3668 {% handling pars
```

See if already started a lateximage or a :

```
3669     \ifthenelse{%
3670         \cnttest{\value{LWR@lateximagedepth}}{>}{0} \OR%
3671         \cnttest{\value{LWR@spandepth}}{>}{0}%
3672     }%
```

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

```
3673     {% no nested par tags
3674         \ifthenelse{%
3675             \cnttest{\value{LWR@spandepth}}{>}{0}%
3676             \AND%
3677             \cnttest{\value{LWR@lateximagedepth}}{=} {0}%
3678         }%
3679         {\ifbool{LWR@intabularmetadata}{\unskip\LWR@htmltagc{br /}}}%
3680         }%
3681     }% no nested par tags
```

If have not already started a lateximage or a :

```
3682     {% yes nest par tags
```

Print a closing tag and some extra vertical space:

```
3683         \unskip%
3684         \LWR@htmltagc{/\LWR@tagregularparagraph}%
3685         \LWR@orignewline%
```

No longer doing a paragraph:

```
3686         \global\boolfalse{LWR@doingapar}%
3687 % Disable the special \env{minipage} \& \cs{hspace} interaction
3688 % until a new minipage is found:
3689 %     \begin{macrocode}
3690         \global\boolfalse{LWR@minipagethispar}%
3691     }% end of yes nest par tags
3692 }% end of handling pars
```

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

```
3693 {% not handling pars
3694     \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3695     {\ifbool{LWR@intabularmetadata}{}{\unskip\LWR@htmltagc{br /}}}%
3696     {}%
3697 }% not handling pars
```

Finish with regular paragraph processing

```
3698 \LWR@origpar%
3699 }
```

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

41 Paragraph start/stop handling

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

Section 40 has the commands which actually generate the tags.

The `everyhook` package is used to generate the opening paragraph tags. The closing tags are generated by `\par`.

for HTML output: 3701 `\begin{warpHTML}`

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

```
3702 \newcommand*{\LWR@startpars}%
3703 {%
```

Ignore if inside a span:

```
3704 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
3705 {}%
3706 {%
```

See if currently handling HTML paragraphs:

```
3707     \ifbool{LWR@doingstartpars}%
```

If already in paragraph mode, do nothing.

```
3708     {%
```

If not currently in paragraph mode:

```
3709   {%
```

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

```
3710       \PushPreHook{par}{\LWR@openparagraph}%
```

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

```
3711       \let\par\LWR@closeparagraph
```

```
3712
```

```
3713   }% an intentionally blank line
```

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

```
3714   \global\setbool{LWR@doingstartpars}{true}%
```

No <par> tag yet to undo:

```
3715   \global\boolfalse{LWR@doingapar}%
```

```
3716 }% nestspan
```

```
3717 }
```

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

```
3718 \newcommand*{\LWR@stoppars}%
```

```
3719 {%
```

Ignore if inside a span:

```
3720 \ifthenelse{\cnttest{\value{LWR@spandepth}}{>}{0}}%
```

```
3721 {}%
```

```
3722 {%
```

See if currently handling HTML paragraphs:

```
3723   \ifbool{LWR@doingapar}%
```

if currently in an HTML paragraph:

```
3724   {%
```

Print a closing tag:

```

3725     \unskip%
3726     \LWR@htmltagc{/\LWR@tagregularparagraph}%
3727     \LWR@orignewline%

```

No longer have an open HTML paragraph:

```

3728     \global\boolfalse{LWR@doingapar}%

```

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

```

3729     \global\boolfalse{LWR@minipagethispar}
3730
3731     }% an intentionally blank line

```

If was not in an HTML paragraph:

```

3732     {}%

```

See if currently allowing HTML paragraphs:

```

3733     \ifbool{LWR@doingstartpars}%

```

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

```

3734     {\ClearPreHook{par}}%

```

Else: do nothing

```

3735     {}%

```

no longer in paragraph mode

```

3736     \global\setbool{LWR@doingstartpars}{false}%

```

no <p> tag to undo:

```

3737     \global\boolfalse{LWR@doingapar}%
3738 }% nestspan
3739 }

```

```

3740 \end{warpHTML}

```

42 Page headers and footers

for HTML & PRINT: 3741 \begin{warpall}

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

```

3742 \newcommand{\LWR@firstpagetop}{} % for the home page alone
3743 \newcommand{\LWR@pagetop}{} % for all other pages
3744 \newcommand{\LWR@pagebottom}{}
3745
3746 \newcommand{\LWR@setfirstpagetop}[1]{%
3747 \renewcommand{\LWR@firstpagetop}{#1}
3748 \catcode'\_ =8
3749 }
```

`\HTMLFirstPageTop` $\{ \langle \textit>text and logos \rangle \}$

```

3750 \newcommand{\HTMLFirstPageTop}{%
3751 \catcode'\_ =12
3752 \LWR@setfirstpagetop
3753 }
```

```

3754 \newcommand{\LWR@setpagetop}[1]{%
3755 \renewcommand{\LWR@pagetop}{#1}
3756 \catcode'\_ =8
3757 }
```

`\HTMLPageTop` $\{ \langle \textit>text and logos \rangle \}$

```

3758 \newcommand{\HTMLPageTop}{%
3759 \catcode'\_ =12
3760 \LWR@setpagetop
3761 }
```

```

3762 \newcommand{\LWR@setpagebottom}[1]{%
3763 \renewcommand{\LWR@pagebottom}{#1}
3764 \catcode'\_ =8
3765 }
```

`\HTMLPageBottom` $\{ \langle \textit>text and logos \rangle \}$

```

3766 \newcommand{\HTMLPageBottom}{%
3767 \catcode'\_ =12
3768 \LWR@setpagebottom
3769 }
```

```

3770 \end{warpall}
```

43 CSS

for HTML output: 3771 `\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.

```
3772 \newcommand*{\LWR@currentcss}{lwarp.css}
```

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

```
3773 \newcommand*{\LWR@newcssb}[1]{%
3774 \renewcommand*{\LWR@currentcss}{#1}
3775 \catcode'\_ =8
3776 }
3777
3778 \newcommand*{\CSSFilename}{
3779 \catcode'\_ =12
3780 \LWR@newcssb
3781 }
3782 \end{warpHTML}
```

for PRINT output: 3783 `\begin{warpprint}`
 3784 `\newcommand*{\CSSFilename}[1]{}`
 3785 `\end{warpprint}`

44 HTML meta description and author

for HTML & PRINT: 3786 `\begin{warpall}`

`\HTMLAuthor` `{\langle authorname \rangle}` The author to place into an HTML meta tag.

```
3787 \providecommand{\theauthor}{}
3788 \newcommand{\theHTMLAuthor}{\theauthor}
3789
3790 \newcommand{\HTMLAuthor}[1]{\renewcommand{\theHTMLAuthor}{#1}}

3791 \end{warpall}
```

for HTML & PRINT: 3792 `\begin{warpall}`

This is placed inside an HTML meta tag at the start of each file. This may be changed mid-document using `\HTMLDescription`, allowing different HTML descriptions to be used for different sections of the document.

 Do not use double quotes, and do not exceed 150 characters.

`\HTMLDescription` `{(New HTML meta description.)}` Assigns the HTML file's description meta tag.

```
3793 \newcommand{\LWR@currentHTMLDescription}{}
3794
3795 \newcommand{\HTMLDescription}[1]{%
3796 \renewcommand{\LWR@currentHTMLDescription}{#1}
3797 }
3798
3799 \end{warpall}
```

45 Footnotes

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

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

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

45.1 Regular page footnotes

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

See section 45.4 for the implementation.

45.2 Minipage footnotes

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

45.3 Titlepage thanks

See section 52.6 for titlepage footnotes.

45.4 Regular page footnote implementation

for HTML output: 3800 `\begin{warpHTML}`

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

3801 `\newbox\LWR@footnotes`

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

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

3802 `\long\def\@makefntext#1{\@thefnmark #1}`

`\@makefnmark`

3803 `\def\@makefnmark{\hbox{\@thefnmark}}`

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

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

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

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

3804 `\long\def\LWR@footnotetext#1{%`

3805 `\global\setbox\LWR@footnotes=\vbox{%`

Add to any current footnotes:

3806 `\unvbox\LWR@footnotes%`

Remember the footnote number for `\ref`:

```

3807 \protected@edef\@currentlabel{%
3808     \csname p@footnote\endcsname\@thefnmark%
3809 }% \@currentlabel

```

Open a group:

```
3810 \color@begingroup%
```

Use HTML superscripts even inside a lateximage:

```
3811 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{##1}}%
```

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

```

3812 \ifthenelse{%
3813     \boolean{LWR@doingstartpars} \AND%
3814     \cnttest{\value{LWR@lateximagedepth}}{=} {0}%
3815 }%
3816 {}%
3817 {\LWR@htmltagc{\LWR@tagregularparagraph}}%

```

Append the footnote to the list:

```
3818 \@makefntext{#1}%
```

Closing paragraph tag:

```

3819 \ifthenelse{%
3820     \boolean{LWR@doingstartpars} \AND%
3821     \cnttest{\value{LWR@lateximagedepth}}{=} {0}%
3822 }%
3823 {\par}%
3824 {%
3825     \LWR@htmltagc{/\LWR@tagregularparagraph}%
3826     \LWR@orignewline%
3827 }%

```

Close the group:

```

3828 \color@endgroup%
3829 }% vbox
3830 }%

```

`\@footnotetext` `{\text}`

```
3831 \let\@footnotetext\LWR@footnotetext
```

45.5 Minipage footnote implementation

`\@mpfootnotetext` $\langle text \rangle$

```

3832 \long\def\@mpfootnotetext#1{%
3833 \global\setbox\@mpfootins\vbox{%
3834   \unvbox\@mpfootins
3835   \reset@font\footnotesize
3836   \hsize\columnwidth
3837   \@parboxrestore
3838   \protected@edef\@currentlabel
3839   {\csname p@mpfootnote\endcsname\@thefnmark}%
3840   \color@begingroup

```

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

```

3841   \ifthenelse{%
3842     \boolean{LWR@doingstartpars} \AND%
3843     \cnttest{\value{LWR@lateximagedepth}}{=}{0}%
3844   }%
3845   {}%
3846   {\LWR@htmltagc{\LWR@tagregularparagraph}}%

3847   \@makefnstext{%
3848     \ignorespaces#1%
3849   }%

```

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

```

3850   \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
3851   {}%
3852   {%
3853     \LWR@htmltagc{/\LWR@tagregularparagraph}%
3854     \LWR@orignewline%
3855   }%
3856   \color@endgroup%
3857 }% vbox
3858 }

```

45.6 Printing pending footnotes

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

```

3859 \newcommand*\LWR@printpendingfootnotes}{%
3860 \ifvoid\LWR@footnotes\else

```

```

3861 \LWR@forcenewpage
3862 \begin{BlockClass}{footnotes}
3863 \LWR@origmedskip
3864 \unvbox\LWR@footnotes
3865 \setbox\LWR@footnotes=\vbox{}
3866 \end{BlockClass}
3867 \fi
3868 }

```

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

```

3869 \newcommand*\LWR@epubprintpendingfootnotes}{%
3870 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}{%
3871 {\LWR@printpendingfootnotes}%
3872 {}}%
3873 }

3874 \end{warpHTML}

```

46 Marginpars

for HTML output: 3875 `\begin{warpHTML}`

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

```

3876 \renewcommand{\marginpar}[2] [] {%
3877 \LWR@htmlspanclass{marginpar}{#2}%
3878 }

3879 \end{warpHTML}

```

47 Splitting HTML files

- Files are split according to `FileDepth` and `CombineHigherDepths`.
- Filenames are sanitized by `\LWR@filenamenooblanks`.
- `\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: 3880 `\begin{warpall}`

Ctrl FileDepth `{<section depth>}` determines how deeply to break into new HTML files, similar to `tocdepth`. The default of `-5` produces one large HTML file.

```
3881 \newcounter{FileDepth}
3882 \setcounter{FileDepth}{-5}
```

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

```
3883 \newbool{CombineHigherDepths}
3884 \booltrue{CombineHigherDepths}
```

```
3885 \end{warpall}
```

for HTML output: 3886 `\begin{warpHTML}`

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

```
3887 \newcommand*{\LWR@thisfilename}{}
```

`\LWR@thisnewfilename` The filename being sanitized.

```
3888 \newcommand*{\LWR@thisnewfilename}{}
```

`\LWR@filenameno blanks` `{<filename>}`

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



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

```
3889 \newcommand*{\LWR@filenameno blanks}[1]{%
3890 \begingroup
```

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

```
3891 \LWR@nullfonts
3892 \renewcommand*{\LWR@htmltagc}[1]{%
```

Replaces common symbols and short words with hyphens:

```
3893 \edef\LWR@thisnewfilename{#1}
3894 \fullexpandarg
```

Convert spaces into hyphens:

```
3895 \StrSubstitute{\LWR@thisnewfilename}{ }{-}[\LWR@thisnewfilename]
```

Convert punctutation into hyphens:

```
3896 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3897 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
3898 \StrSubstitute{\LWR@thisnewfilename}%
3899 {\LWR@origampersand}{-}[\LWR@thisnewfilename]
3900 \StrSubstitute{\LWR@thisnewfilename}{+}{-}[\LWR@thisnewfilename]
3901 \StrSubstitute{\LWR@thisnewfilename}{,}{-}[\LWR@thisnewfilename]
3902 \StrSubstitute{\LWR@thisnewfilename}{/}{-}[\LWR@thisnewfilename]
3903 \StrSubstitute{\LWR@thisnewfilename}{:}{-}[\LWR@thisnewfilename]
3904 \StrSubstitute{\LWR@thisnewfilename}{;}{-}[\LWR@thisnewfilename]
3905 \StrSubstitute{\LWR@thisnewfilename}{=}{-}[\LWR@thisnewfilename]
3906 \StrSubstitute{\LWR@thisnewfilename}{?}{-}[\LWR@thisnewfilename]
3907 \StrSubstitute{\LWR@thisnewfilename}{@}{-}[\LWR@thisnewfilename]
3908 \StrSubstitute{\LWR@thisnewfilename}{"}{-}[\LWR@thisnewfilename]
3909 \StrSubstitute{\LWR@thisnewfilename}%
3910 {\textless}{-}[\LWR@thisnewfilename]
3911 \StrSubstitute{\LWR@thisnewfilename}%
3912 {\textgreater}{-}[\LWR@thisnewfilename]
3913 \StrSubstitute{\LWR@thisnewfilename}{\#}{-}[\LWR@thisnewfilename]
3914 \StrSubstitute{\LWR@thisnewfilename}{\%}{-}[\LWR@thisnewfilename]
3915 \StrSubstitute{\LWR@thisnewfilename}{\{}{-}[\LWR@thisnewfilename]
3916 \StrSubstitute{\LWR@thisnewfilename}{\}}{-}[\LWR@thisnewfilename]
3917 \StrSubstitute{\LWR@thisnewfilename}{|}{-}[\LWR@thisnewfilename]
3918 \StrSubstitute{\LWR@thisnewfilename}%
3919 {\textbackslash}{-}[\LWR@thisnewfilename]
3920 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3921 \StrSubstitute{\LWR@thisnewfilename}{~}{-}[\LWR@thisnewfilename]
3922 %    "~{" for babel
3923 \StrSubstitute{\LWR@thisnewfilename}{[}{-}[\LWR@thisnewfilename]
3924 \StrSubstitute{\LWR@thisnewfilename}{]}{-}[\LWR@thisnewfilename]
3925 \StrSubstitute{\LWR@thisnewfilename}{'}{-}[\LWR@thisnewfilename]
```

Convert short words:

```
3926 \StrSubstitute{\LWR@thisnewfilename}{-s-}{-}[\LWR@thisnewfilename]
3927 \StrSubstitute{\LWR@thisnewfilename}{-S-}{-}[\LWR@thisnewfilename]
3928 \StrSubstitute{\LWR@thisnewfilename}{-a-}{-}[\LWR@thisnewfilename]
3929 \StrSubstitute{\LWR@thisnewfilename}{-A-}{-}[\LWR@thisnewfilename]
3930 \StrSubstitute{\LWR@thisnewfilename}{-an-}{-}[\LWR@thisnewfilename]
3931 \StrSubstitute{\LWR@thisnewfilename}{-AN-}{-}[\LWR@thisnewfilename]
3932 \StrSubstitute{\LWR@thisnewfilename}{-to-}{-}[\LWR@thisnewfilename]
3933 \StrSubstitute{\LWR@thisnewfilename}{-TO-}{-}[\LWR@thisnewfilename]
3934 \StrSubstitute{\LWR@thisnewfilename}{-by-}{-}[\LWR@thisnewfilename]
3935 \StrSubstitute{\LWR@thisnewfilename}{-BY-}{-}[\LWR@thisnewfilename]
```

```

3936 \StrSubstitute{\LWR@thisnewfilename}{-of-}{-}[\LWR@thisnewfilename]
3937 \StrSubstitute{\LWR@thisnewfilename}{-OF-}{-}[\LWR@thisnewfilename]
3938 \StrSubstitute{\LWR@thisnewfilename}{-and-}{-}[\LWR@thisnewfilename]
3939 \StrSubstitute{\LWR@thisnewfilename}{-AND-}{-}[\LWR@thisnewfilename]
3940 \StrSubstitute{\LWR@thisnewfilename}{-for-}{-}[\LWR@thisnewfilename]
3941 \StrSubstitute{\LWR@thisnewfilename}{-FOR-}{-}[\LWR@thisnewfilename]
3942 \StrSubstitute{\LWR@thisnewfilename}{-the-}{-}[\LWR@thisnewfilename]
3943 \StrSubstitute{\LWR@thisnewfilename}{-THE-}{-}[\LWR@thisnewfilename]

```

Convert multiple hyphens:

```

3944 \StrSubstitute{\LWR@thisnewfilename}{-----}{-}[\LWR@thisnewfilename]
3945 \StrSubstitute{\LWR@thisnewfilename}{----}{-}[\LWR@thisnewfilename]
3946 \StrSubstitute{\LWR@thisnewfilename}{---}{-}[\LWR@thisnewfilename]
3947 \StrSubstitute{\LWR@thisnewfilename}{--}{-}[\LWR@thisnewfilename]
3948 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3949 %      emdash
3950 \StrSubstitute{\LWR@thisnewfilename}{-}{-}[\LWR@thisnewfilename]
3951 %      endash
3952 \global\let\LWR@thisfilename\LWR@thisnewfilename% return a global result
3953 \endgroup
3954 }

```

`\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.

```

3955 \newcommand*{\LWR@newhtmlfile}[1]{

```

At the bottom of the ending file:

```

3956 \LWR@htmlElementclassend{section}{tbody}
3957
3958 \LWR@printpendingfootnotes
3959

```

No footer between files if EPUB:

```

3960 \ifbool{FormatEPUB}
3961 {}
3962 {
3963   \LWR@htmlElement{footer}
3964
3965   \LWR@pagebottom
3966

```

```
3967 \LWR@html elementend{footer}
3968 }
```

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

```
3969 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3970 {}
3971 {\ifnumcomp{\value{LWR@htmlfilename}}{>}{0}{\LWR@botnavigation}{}}
```

End of this HTML file:

```
3972 \LWR@stoppars
3973 \LWR@htmltag{/body}\LWR@orignewline
3974 \LWR@htmltag{/html}\LWR@orignewline
3975 \LWR@orignewpage
3976
3977 \addtocounter{LWR@htmlfilename}{1}%
```

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

```
3978 \ifbool{FileSectionNames}%
3979 {\LWR@filenameno blanks{#1}}
3980 {\renewcommand*{\LWR@thisfilename}{\theLWR@htmlfilename}}
```

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.

```
3981 \LWR@htmlblockcomment{%
3982 |Start file|
3983 \LWR@htmlsectionfilename{\LWR@thisfilename}|
3984 }
```

At the top of the starting file:

```
3985 \LWR@stoppars
3986
3987 \LWR@filestart{ -- #1}% there is an EMDash in front of the #1
3988
```

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

```
3989 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
```

```
3990 {}
3991 {\LWR@topnavigation}
3992
```

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

```
3993 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
3994 {}
3995 {
3996   \LWR@htmlElement{header}
3997
3998   \LWR@pagetop
3999
4000   \LWR@htmlElementend{header}
4001 }
4002
```

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

```
4003 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4004 {}
4005 {\ifcvoid{thetitle}{}\LWR@printthetitle}
4006
```

No sidetoc if formatting for an EPUB or word processor:

```
4007 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4008 {}
4009 {\LWR@sidetoc}
4010
```

Start of the <textbody>:

```
4011 \LWR@htmlElementclass{section}{textbody}
4012
```

Keep paragraph tags disabled for now:

```
4013 \LWR@stoppars
4014
```

Track the page numbers:

```
4015 \setcounter{LWR@latestautopage}{\value{page}}
4016 }
```

```
4017 \end{warpHTML}
```

48 Sectioning

Sectioning and cross-references have been emulated from scratch, rather than try to patch several layers of existing \TeX code and packages. Formatting is handled by CSS, so the emulated code has much less work to do than the print versions.

Unicode Section names and the resulting filenames with accented characters are partially supported, depending on the ability of `pdf \TeX` to generate characters and `pdftotext` to read them. If extra symbols appear in the text, it may be that `pdf \TeX` is actually producing a symbol over or under a character, resulting in `pdftotext` picking up the accent symbol separately.



$\Xe $\TeX$$ and $\text{Lua}\mathbb{X}$ directly support accented section and file names.

for HTML output: `4018 \begin{warpHTML}`

48.1 User-level starred section commands

`\ForceHTMLPage` For HTML output, forces the next section to be on its own HTML page, if `FileDepth` allows, even if starred. For use with `\printindex` and others which generate a starred section which should be on its own HTML page. Also see `\ForceHTMLTOC`.

For print output, no effect.

```
4019 \newbool{LWR@forcinghtmlpage}
4020 \boolfalse{LWR@forcinghtmlpage}
4021
4022 \newcommand*{\ForceHTMLPage}{%
4023 \global\booltrue{LWR@forcinghtmlpage}%
4024 }
```

`\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.

```
4025 \newbool{LWR@forcinghtmltoc}
4026 \boolfalse{LWR@forcinghtmltoc}
4027
4028 \newcommand*{\ForceHTMLTOC}{%
4029 \global\booltrue{LWR@forcinghtmltoc}%
4030 }
```

```
4031 \end{warpHTML}
```

for PRINT output:

```
4032 \begin{warpprint}
4033 \newcommand*{\ForceHTMLPage}{}
4034 \newcommand*{\ForceHTMLTOC}{}
4035 \end{warpprint}
```

for HTML output:

```
4036 \begin{warpHTML}
```

48.2 Book class commands

`\mainmatter`  Declare the main matter section of the document. Does not reset the page number, which must be consecutive arabic numbers for the HTML conversion.

```
4037 \newbool{LWR@mainmatter}
4038 \DeclareDocumentCommand{\mainmatter}{}{}%
4039 \booltrue{LWR@mainmatter}%
4040 }
```

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

```
4041 \DeclareDocumentCommand{\frontmatter}{}{}%
4042 \boolfalse{LWR@mainmatter}%
4043 }
```

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

```
4044 \DeclareDocumentCommand{\backmatter}{}{}%
4045 \boolfalse{LWR@mainmatter}
4046 }
```

48.3 Sectioning support macros

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

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

```
4047 \newcommand*{\LWR@sectionnumber}[1]{}%
4048 \InlineClass{sectionnumber}{{#1}}%
4049 }
```

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

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

Create an autosection tag.

```
4050 \newcommand*\LWR@createautosec[1]{%
4051 \LWR@htmltag{#1 id="autosec-\thepage"{} }%
4052 }
```

`\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.

```
4053 \NewDocumentCommand{\LWR@pushoneclose}{m m}{\pushclose{#2}{#1}}
```

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

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

 `\LWR@stoppars` must be executed first.

```
4054 \NewDocumentCommand{\LWR@startnewdepth}{m m}{%
```

Close any stacked sections up to this new one.

```
4055 \LWR@closeprevious{#1} %
```

Push a new section depth:

```
4056 \LWR@pushoneclose{#1}{#2} %
4057 }
```

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.

```
4058 \newcounter{LWR@prevFileDepth}
4059 \setcounter{LWR@prevFileDepth}{\LWR@depthsubparagraph}
```

`\@secntformat` $\{\langle sectiontype \rangle\}$

```
4060 \def\@secntformat#1{\csname the#1\endcsname\protect\quad}
```

`\simplechapterdelim` Used by `tocbibind` and `anonchap`.

```
4061 \newcommand*\simplechapterdelim{}
```

`\@chapcntformat` $\langle sectiontype \rangle$

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

```
4062 \let\@chapcntformat\@secntformat
```

`\LWR@section` * $[(TOC\ name)] \langle name \rangle \langle sectiontype \rangle$

The common actions for the high-level sectioning commands.

```
4063 \DeclareDocumentCommand\LWR@section{m m m m}{%
4064 \LWR@traceinfo{LWR@section |#2| |#3|}%
4065 % \ifthenelse{%
4066 % % \equal{#2}{\NoValue-}\AND%
4067 % \isempty{#3}%
4068 % }%
4069 % {% empty section name
4070 % \LWR@traceinfo{LWR@section: empty section}%
4071 % }% empty section name
4072 % {% not an empty section name
4073 \LWR@traceinfo{LWR@section: not an empty section}%
4074 \LWR@stoppars%
```

Cancel special minipage horizontal space interaction:

```
4075 \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:

```
4076 \LWR@traceinfo{LWR@section: testing whether to start a new HTML file}%
4077 \IfBooleanTF{#1}{\LWR@traceinfo{LWR@section: starred}}{%
4078 \ifbool{LWR@forcinghtmlpage}{\LWR@traceinfo{LWR@section: forcinghtmlpage}}{%
4079 \ifthenelse{%
4080   \(%
4081     \(\NOT\equal{#1}{\BooleanTrue}\)\OR%
4082     \(\cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}\)\OR%
4083     \(\boolean{LWR@forcinghtmlpage}\)\%
4084   \)%
4085   \AND%
```

```

4086 \cntttest{\csuse{LWR@depth#4}}{<=}{\value{FileDepth}}%
4087 \AND%
4088 \(%
4089 \NOT\boolean{CombineHigherDepths}\OR%
4090 \cntttest{\csuse{LWR@depth#4}}{<=}{\value{LWR@prevFileDepth}}%
4091 \)%
4092 \AND%
4093 \NOT\equal{#3}{}% phantomsection
4094 }%

```

If so: start a new HTML file:

```

4095 {% new file
4096 \LWR@traceinfo{LWR@section: new HTML file}%

```

See if there was an optional TOC name entry:

```

4097 \IfNoValueTF{#2}%

```

If no optional entry

```

4098 {\LWR@newhtmlfile{#3}}%

```

If yes an optional entry

```

4099 {\LWR@newhtmlfile{#2}}%
4100 }% new file

```

Else: No new HTML file:

```

4101 {% not new file

```

Generate a new \LaTeX page so that toc and index page number points to the section:

```

4102 \LWR@traceinfo{LWR@section: not a new HTML file}%
4103 \LWR@orignewpage%
4104
4105 }% not new file

```

Remember this section's name for \nameref:

```

4106 \IfValueT{#3}{%
4107 \LWR@traceinfo{LWR@section: about to LWR@setlatestname}%
4108 \IfValueTF{#2}{\LWR@setlatestname{#2}}{\LWR@setlatestname{#3}}%
4109 }%

```

Print an opening comment with the level and the name; ex: "section" "Introduction"

```

4110
4111 \ifbool{HTMLDebugComments}{%
4112   \LWR@htmlcomment{Opening #4 ‘#3’}{}}%
4113 }{}%
4114

```

For inline sections paragraph and subparagraph, start a new paragraph now:

```

4115 \ifthenelse{%
4116   \cnttest{\csuse{LWR@depth#4}}{>=}{\LWR@depthparagraph}%
4117 }%
4118 {\LWR@startpars}%
4119 {}%

```

Create the opening tag with an autosec:

```

4120 \LWR@createautosec{\csuse{LWR@tag#4}}%

```

Check if starred:

```

4121 \IfBooleanTF{#1}%
4122 {%

```

Starred, but also forcing a TOC entry, so add unnumbered TOC name or regular name:

```

4123 \ifbool{LWR@forcinghtmltoc}%
4124 {\addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}}%
4125 }%
4126 }% starred

```

Not starred, so step counter and add to TOC:

```

4127 {% not starred

```

Only add a numbered TOC entry if section number is not too deep:

```

4128   \ifthenelse{%
4129     \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4130   }%
4131   {% if secnumdepth

```

If in the main matter, step the counter and add the TOC entry. For article class, lwarp assumes that all is mainmatter.

```

4132     \LWR@traceinfo{LWR@section: about to test main matter}%
4133     \ifbool{LWR@mainmatter}%
4134     {%

```

```
4135         \LWR@traceinfo{LWR@section: yes mainmatter}%
4136         \refstepcounter{#4}%
```

Add main matter numbered TOC entry with the TOC name or the regular name:

```
4137         \LWR@traceinfo{LWR@section: about to addcontentsline}%
4138         \addcontentsline{toc}{#4}%
4139         {%
4140             \protect\numberline{\csuse{the#4}}%
4141             {\ignorespaces\IfValueTF{#2}{#2}{#3}\protect\relax}%
4142         }%
4143         \LWR@traceinfo{LWR@section: finished addcontentsline}%
4144     }% end of if main matter
```

If not main matter, add unnumbered TOC name or regular name:

```
4145     {% not main matter
4146         \LWR@traceinfo{LWR@section: no main matter}%
4147         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4148     }% end of not main matter
4149 }% end of secnumdepth
```

Deeper than secnumdepth, so add an unnumbered TOC entry:

```
4150     {%
4151         \addcontentsline{toc}{#4}{\IfValueTF{#2}{#2}{#3}}%
4152     }%
```

For part, print the section type:

```
4153     \ifbool{LWR@mainmatter}%
4154     {%
4155         \ifthenelse{%
4156             \(\cnttest{\csuse{LWR@depth#4}}{<=}%
4157             {\value{secnumdepth}}\)\ AND%
4158             \(\cnttest{\csuse{LWR@depth#4}}{<=}{\LWR@depthpart}\)%
4159         }%
4160         {\csuse{#4name}~{}}%
4161     }%
```

Print the section number:

```
4162         \LWR@traceinfo{LWR@section: about to print section number}%
4163         \ifthenelse{%
4164             \cnttest{\csuse{LWR@depth#4}}{<=}{\value{secnumdepth}}%
4165         }%
4166         {%
4167             \ifstrequal{#4}{chapter}%
4168             {\protect\LWR@sectionnumber{\@chapcntformat{#4}}}%

```

```

4169         {\protect\LWR@sectionnumber{\@secntformat{#4}}}%
4170     }%
4171     {}%
4172     \LWR@traceinfo{LWR@section: finished print section number}%
4173     }{}%
4174 }% end of not starred

```

Print the section name:

```
4175 #3%
```

Close the heading tag, such as /H2:

```
4176 \LWR@htmltag{\csuse{LWR@tag#4end}}%
```

Generate a \LaTeX label:

```
4177 \label{autopage-\thepage}%
```

Start paragraph handling unless is an inline paragraph or subparagraph:

```

4178 \ifthenelse{%
4179     \cnttest{\csuse{LWR@depth#4}}{<}{\LWR@depthparagraph}%
4180 }%
4181 {\LWR@startpars}%
4182 {}%

```

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.

```

4183 \ifthenelse{%
4184     \NOT\equal{#1}{\BooleanTrue}\OR%
4185     \cnttest{\csuse{LWR@depth#4}}{=}{\LWR@depthpart}%
4186 }%
4187 {% not starred
4188     \setcounter{LWR@prevFileDepth}{\csuse{LWR@depth#4}}%
4189 }% not starred
4190 {}%

```

Reset to defaults if not a phantomsection:

```

4191 \ifstrempy{#3}%
4192 {}%
4193 {%
4194 \global\boolfalse{LWR@forcinghtmlpage}%
4195 \global\boolfalse{LWR@forcinghtmltoc}%
4196 }%
4197 %
4198 % }% not an empty section name
4199 \LWR@traceinfo{LWR@section: done}%
4200 }

```

48.4 \section and friends

`\part` * [*TOC name*] {*name*}

```

4201 \DeclareDocumentCommand{\part}{s o m}{%
4202 \LWR@epubprintpendingfootnotes%
4203 \LWR@stoppars%
4204
4205 \LWR@startnewdepth{\LWR@depthpart}{\LWR@printclosepart}%
4206
4207 \LWR@section{#1}{#2}{#3}{part}%
4208 }

```

`\chapter` * [*TOC name*] {*name*}

```

4209 \let\@printcites\relax% for quotchap package
4210
4211 \ifundefined{chapter}
4212 {}
4213 {%
4214 \DeclareDocumentCommand{\chapter}{s o m}{%
4215 \LWR@traceinfo{chapter #3}%
4216 \LWR@epubprintpendingfootnotes%
4217 \LWR@stoppars%
4218
4219 \LWR@startnewdepth{\LWR@depthchapter}{\LWR@printclosechapter}%
4220
4221 \LWR@section{#1}{#2}{#3}{chapter}%
4222 \@printcites% for quotchap package
4223 }
4224 }

```

`\section` * [*TOC name*] {*name*}

```

4225 \DeclareDocumentCommand{\section}{s o m}{%
4226 \LWR@epubprintpendingfootnotes%
4227 \LWR@stoppars%
4228
4229 \LWR@startnewdepth{\LWR@depthsection}{\LWR@printclosesection}%
4230
4231 \LWR@section{#1}{#2}{#3}{section}%
4232 }

```

`\subsection` * [*TOC name*] {*name*}

```

4233 \DeclareDocumentCommand{\subsection}{s o m}{%
4234 \LWR@epubprintpendingfootnotes%
4235 \LWR@stoppars%
4236
4237 \LWR@startnewdepth{\LWR@depthsubsection}{\LWR@printclosesubsection}%
4238
4239 \LWR@section{#1}{#2}{#3}{subsection}%
4240 }

```

`\subsubsection` * [*TOC name*] {*name*}

```

4241 \DeclareDocumentCommand{\subsubsection}{s o m}{%
4242 \LWR@epubprintpendingfootnotes%
4243 \LWR@stoppars%
4244
4245 \LWR@startnewdepth{\LWR@depthsubsubsection}{%
4246 {\LWR@printclosesubsubsection}}%
4247
4248 \LWR@section{#1}{#2}{#3}{subsubsection}%
4249 }

```

`\paragraph` * [*TOC name*] {*name*}

```

4250 \DeclareDocumentCommand{\paragraph}{s o m}{%
4251 \LWR@epubprintpendingfootnotes%
4252 \LWR@stoppars%
4253
4254 \LWR@startnewdepth{\LWR@depthparagraph}{\LWR@printcloseparagraph}%
4255
4256 \LWR@section{#1}{#2}{#3}{paragraph}%
4257 }

```

`\subparagraph` * [*TOC name*] {*name*}

```

4258 \DeclareDocumentCommand{\subparagraph}{s o m}{%
4259 \LWR@epubprintpendingfootnotes%

```

```

4260 \LWR@stoppars%
4261
4262 \LWR@startnewdepth{\LWR@depthsubparagraph}{\LWR@printclosesubparagraph}%
4263
4264 \LWR@section{#1}{#2}{#3}{subparagraph}%
4265 }

4266 \end{warpHTML}

```

49 Starting a new file

for HTML & PRINT: 4267 \begin{warpall}

\HTMLLanguage Default language for the HTML lang tag.

```

4268 \newcommand*\LWR@currentHTMLLanguage{en-US}
4269
4270 \newcommand*\HTMLLanguage[1]{%
4271 \renewcommand*\LWR@currentHTMLLanguage{#1}%
4272 }

4273 \end{warpall}

```

for HTML output: 4274 \begin{warpHTML}

\LWR@filestart $\{ \langle title_suffix \rangle \}$

Creates the opening HTML tags.

```

4275 \newcommand*\LWR@filestart[1]{
4276 \LWR@traceinfo{LWR@filestart}

```

Locally temporarily disable direct-formatting commands:

```

4277 \begingroup
4278 \LWR@traceinfo{LWR@filestart: A}
4279 \LWR@nullfonts
4280 \LWR@traceinfo{LWR@filestart: B}

```

Create the page's HTML header:

```

4281 \LWR@htmltag{!DOCTYPE html}\LWR@orignewline
4282 \LWR@traceinfo{LWR@filestart: C}

```

The language is user-adjustable:

```
4283 \LWR@htmltag{html lang="\LWR@currentHTMLLanguage"}\LWR@orignewline
```

Start of the meta data:

```
4284 \LWR@htmltag{head}\LWR@orignewline
```

Charset is fixed at UTF-8:

```
4285 \LWR@htmltag{meta charset="UTF-8" /}\LWR@orignewline
```

Author:

```
4286 \ifcsemtyp{theHTMLAuthor}{}{  
4287 \LWR@htmltag{meta name="author" content="\theHTMLAuthor" /}\LWR@orignewline  
4288 }
```

lwarp is the generator:

```
4289 \LWR@htmltag{meta name="generator" content="LaTeX lwarp package" /}%  
4290 \LWR@orignewline
```

If there is a description, add it now:

```
4291 \ifdefempty{\LWR@currentHTMLDescription}{}{%  
4292 \LWR@htmltag{%  
4293 meta name="description" content="\LWR@currentHTMLDescription" /}%  
4294 \LWR@orignewline  
4295 }%
```

Mobile-friendly viewport:

```
4296 \LWR@htmltag{meta name="viewport" %  
4297 content="width=device-width, initial-scale=1.0" /}%  
4298 \LWR@orignewline
```

IE patch:

```
4299 \LWR@htmltag{!{-}{-}[if lt IE 9]}\LWR@orignewline  
4300 \LWR@htmltag{%  
4301 script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"}%  
4302 \LWR@htmltag{/script}\LWR@orignewline  
4303 \LWR@htmltag{![endif]{-}{-}}\LWR@orignewline
```

The page's title:

```
4304 \ifcvoid{thetitle}{}{%}
```

```
4305 \LWR@htmltag{title}\thetitle#1\LWR@htmltag{/title}\LWR@orignewline%
4306 }%
```

The page's stylesheet:

```
4307 \LWR@htmltag{%
4308 link rel="stylesheet" type="text/css" href="\LWR@currentcss" /}%
4309 \LWR@orignewline
```

Optional MathJax support. The HTML tags must be turned off during the verbatim input, and the paragraph handling which was turned on at the end of verbatim input must be immediately turned off again.

```
4310 \ifbool{mathjax}%
4311 {%
4312   \boolfalse{LWR@verbtags}
4313   \verbatiminput{lwarp_mathjax.txt}%
4314   \booltrue{LWR@verbtags}
4315   \LWR@stoppars
4316 }% end of mathjax
4317 {}%
```

End of the header:

```
4318 \LWR@htmltag{/head}\LWR@orignewline
```

Start of the body:

```
4319 \LWR@htmltag{body}\LWR@orignewline
4320 \endgroup
4321 \LWR@traceinfo{LWR@filestart: done}
4322 }

4323 \end{warpHTML}
```

50 Starting HTML output

for HTML output: 4324 \begin{warpHTML}

\LWR@LwarpStart Executed at the beginning of the entire document.

```
4325 \catcode'\$=\active
4326 \newcommand*{\LWR@LwarpStart}
4327 {%
4328 \LWR@traceinfo{LWR@lwarpStart}
```

If formatting for a word processor, force filedepth to single-file only, force HTML debug comments off.

```
4329 \ifbool{FormatWordProcessor}{%  
4330   \setcounter{FileDepth}{-5}%  
4331   \boolfalse{HTMLDebugComments}%  
4332 }{}
```

Expand and detokenize \HomeHTMLFilename and \HTMLFilename:

```
4333 \edef\LWR@strresult{\HomeHTMLFilename}  
4334 \edef\HomeHTMLFilename{\detokenize\expandafter{\LWR@strresult}}  
4335 \edef\LWR@strresult{\HTMLFilename}  
4336 \edef\HTMLFilename{\detokenize\expandafter{\LWR@strresult}}
```

Force onecolumn and empty page style:

```
4337 \LWR@origonecolumn%  
4338 \LWR@origpagestyle{empty}
```

Reduce chance of line overflow in verbatim environments:

```
4339 \LWR@origscriptsize%
```

In PDF output, don't allow line breaks to interfere with HTML tags:

```
4340 \LWR@origraggedright%  
4341 \LetLtxMacro{\}\{\LWR@endofline}%
```

Spread the lines for pdftotext to read them well:

```
4342 \linespread{1.3}%
```

For pdftotext to reliably identify paragraph splits:

```
4343 \setlength{\parindent}{0pt}  
4344 \setlength{\parskip}{2ex}
```

For the lateximages record file:

```
4345 \immediate\openout\LWR@lateximagesfile=lateximages.txt
```

Removes space after the caption in the HTML:

```
4346 \setlength{\belowcaptionskip}{-3ex}
```

Redefine the plain page style to be empty when used by index pages:

```
4347 \renewcommand{\ps@plain}{}

```

```

\centering Not used in the HTML environment:
\raggedleft
\raggedright 4348 \renewcommand*{\centering}{}
4349 \renewcommand*{\raggedleft}{}
4350 \renewcommand*{\raggedright}{}

```

Plug in some new actions. This is done just before the document start so that they won't be over-written by some other package.

Tabular:

```

4351 \LetLtxMacro{\LWR@origtabular}{\tabular}
4352 \LetLtxMacro{\LWR@origendtabular}{\endtabular}
4353 \LetLtxMacro{\tabular}{\LWR@tabular}
4354 \LetLtxMacro{\endtabular}{\endLWR@tabular}

```

Float captions:

```
4355 \let\LWR@origcaption\caption

```

Labels: `\ltx@label` is used in amsmath environments and is also patched by cleveref.

[Label in HTML](#)

```

4356 \let\LWR@origltx@label\ltx@label
4357 \let\ltx@label\LWR@htmlmathlabel

```

Logos:

```

4358 \let\TeX\LWR@TeX
4359 \let\LaTeX\LWR@LaTeX
4360 \let\LuaTeX\LWR@LuaTeX
4361 \let\LuaLaTeX\LWR@LuaLaTeX
4362 \let\XeTeX\LWR@XeTeX
4363 \let\XeLaTeX\LWR@XeLaTeX
4364 \let\ConTeXt\LWR@ConTeXt

```

Not yet started any paragraph handling:

```

4365 \global\boolfalse{LWR@doingapar}
4366 \global\boolfalse{LWR@doingstartpars}

```

Start a new HTML file and a header:

```

4367 \LWR@traceinfo{LWR@lwarpStart: Starting new file.}
4368 \LWR@filestart{}
4369 \LWR@traceinfo{LWR@lwarpStart: Generating first header.}

```

```

4370 \LWR@htmltag{header}\LWR@orignewline
4371 \LWR@startpars
4372 \LWR@firstpagetop
4373 \LWR@stoppars
4374 \LWR@htmltag{/header}\LWR@orignewline
4375 \LWR@traceinfo{LWR@lwarpStart: Generating textbody.}
4376 \LWR@htmltag{section class="textbody"{}

```

Document and page settings:

```

4377 \mainmatter
4378 \LWR@origpagenumbering{arabic}

```

Patch the `itemize`, `enumerate`, and `description` environments and `\item`. This works with the native \TeX environments, as well as those provided by `enumitem`, `enumerate`, and `paralist`.

```

4379 \LWR@patchlists

```

Ensure that math mode is active to call lwarp's patches:

```

4380 \catcode'\$=\active

```

Allow HTML paragraphs to begin:

```

4381 \LWR@startpars
4382 \LWR@traceinfo{LWR@lwarpStart: done}
4383 }
4384 \catcode'\$=3% math shift until lwarp starts
4385 \end{warpHTML}

```

51 Ending HTML output

for HTML output: 4386 `\begin{warpHTML}`

`\LWR@requesttoc` $\{ \langle boolean \rangle \} \{ \langle suffix \rangle \}$ Requests that a toc, lof, or lot be generated.

```

4387 \newcommand*\LWR@requesttoc}[2]{%
4388 \ifbool{#1}
4389 {
4390   \expandafter\newwrite\csuse{tf@#2}
4391   \immediate\openout \csuse{tf@#2} \jobname.#2\relax
4392 }{}
4393 }

```

\LWR@LwarpEnd Final stop of all HTML output:

```
4394 \newcommand*{\LWR@LwarpEnd}
4395 {
4396 \LWR@stoppars
4397 \LWR@closeprevious{\LWR@depthfinished}
```

At the bottom of the ending file:

Close the textbody:

```
4398 \LWR@htmlclassend{section}{textbody}
```

Print any pending footnotes:

```
4399 \LWR@printpendingfootnotes
```

Create the footer:

```
4400 \LWR@htmlclassend{footer}
4401
4402 \LWR@pagebottom
4403
4404 \LWR@htmlclassend{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.

```
4405 \ifthenelse{\boolean{FormatEPUB}\OR\boolean{FormatWordProcessor}}
4406 {}
4407 {
4408   \ifnumcomp{\value{LWR@htmlfilenumber}}{>}{0}{\LWR@botnavigation}{}
4409 }
```

```
4410 \LWR@stoppars% final stop of all paragraphs
```

Finish the HTML file:

```
4411 \LWR@htmltag{/body}\LWR@orignewline
4412 \LWR@htmltag{/html}\LWR@orignewline
```

Seems to be required sometimes:

```
4413 \LWR@orignewpage
```

For lateximage commands:

```
4414 \immediate\closeout\LWR@lateximagesfile
4415 }

4416 \end{warpHTML}
```

52 Title page

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

 **load order**

\published and \subtitle If using the titling package, additional titlepage fields for **\published** and **\subtitle** may be added by using **\AddSubtitlePublished** in the preamble. See section 52.7.

affiliation lwarp provides for the **\author** macro an additional **\affiliation** macro to provide an affiliation and other additional information for each author in the title page. The affiliation information is removed when using **titlingpage's \theauthor** in the main text.

reusing titlepage information The titling package maintains the definitions of **\thetitle**, **\theauthor**, etc., after the title has been typeset. These commands are to be used to refer to the document's title and author, etc., in the main text. These definitions have the **\thanks** and **\affiliation** removed, and for **\author** the **\and** is replaced to generate a simple inline list of authors separated by commas. Note: **\theauthor** does not work well with **authblk** unless the traditional \LaTeX syntax is used.

 **\theauthor, authblk**

custom titlepages **\printtitle**, **\printauthor**, etc., are provided for use inside a custom titlepage or **titlingpage** environment, and these retain the **\thanks** and **\affiliation**.

\printthanks **\printthanks** has been added to force the printing of thanks inside a **titlingpage** environment when **\maketitle** is not used.

 Inside a **\titlepage** or **\titlingpage** environment, use **\thanks** instead of **\footnote** for acknowledgements, etc.

52.1 Setting the title, etc.

The following provide setting commands for both HTML and print outputs.

\author $\langle author \rangle$ While using **\maketitle** and print mode, the author is treated as a single-column tabular and the **\and** feature finishes the current tabular then starts a new one for the next author. Each author thus is placed into its own tabular, and an affiliation may be placed on its own line such as

\and

```
\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`, `\@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 `tabular`s if multiples authors are included. For HTML these `tabular`s become side-by-side `<div>`s of class `oneauthor`, all of which are combined into one `<div>` of class `author`.

`\printtitle` `\printtitle`, etc. are user-level macros intended to be used in custom `titlepage` or `titlingpage` environments in cases where `\maketitle` is not desired. These commands preserve the `\thanks`, etc., and should not be used in the main text.

`\thetitle` `\thetitle`, `\theauthor`, and `\thedata` are available if titling has been loaded, and are sanitized user-level versions from which have been removed the `\thanks` and `\affiliation`, and `\and` is changed for inline text usage. The author is printed inline without `\affiliation` or `\thanks`, with `\and` placing commas between multiple authors. Thus, these commands are to be used in the main text whenever the user wishes to refer to the document's title and such. One practical use for this is to place the authors at the bottom of each HTML page, such as:

```
\HTMLPageBottom {<text>}
```

```
\HTMLPageBottom{
\begin{center}\textcopyright~2016 \theauthor\end{center}
}
```

⚠ `\theauthor` `\theauthor` does not work well if `authblk` is used. If `\theauthor` is important, it is recommended to use the standard \TeX syntax for `\author`, optionally with `lwarp`'s `\affiliation` macro as well.

⚠ `affiliations` After `\maketitle` has completed, `\theauthor` retains the definition of the author, but `\and` is changed to become a comma and a space, intending to print the authors names separated by spaces. This fails when affiliations are included on their own table rows.

`\affiliation` A solution, provide here, is to define a macro `\affiliation` which, during `\maketitle`, starts a new row and adds the affiliation, but after `\maketitle` is finished `\affiliation` is re-defined to discard its argument, thus printing only the author names when `\author` is later used inline.

52.2 Changes for `\affiliation`

`\affiliation` $\langle text \rangle$

Adds the affiliation to the author for use in `\maketitle`.

Inside `titlepage`, this macro prints its argument. Outside, it is null.

for HTML & PRINT: 4417 `\begin{warpall}`
 4418 `\newrobustcmd{\affiliation}[1]{}`
 4419 `\end{warpall}`

for PRINT output: 4420 `\begin{warpprint}`

4421 `\AtBeginEnvironment{titlepage}{`
 4422 `\renewrobustcmd{\affiliation}[1]{\ \ \textsc{\small#1}}`
 4423 `}`
 4424
 4425 `\AtBeginDocument{`
 4426 `\@ifpackageloaded{titling}{`
 4427 `\AtBeginEnvironment{titlingpage}{`
 4428 `\renewrobustcmd{\affiliation}[1]{\ \ \textsc{\small#1}}`
 4429 `}`
 4430 `}% titling loaded`
 4431 `}% AtBeginDocument`

4432 `\end{warpprint}`

for HTML output: 4433 `\begin{warpHTML}`

Env `titlepage` Sets up a `<div>` of class `titlepage`.

4434 `\renewenvironment*{titlepage}`
 4435 `{`
 4436 `\renewrobustcmd{\affiliation}[1]{\ \ \InlineClass{affiliation}{##1}}`
 4437 `\LWR@printpendingfootnotes`
 4438 `\LWR@forcenewpage`
 4439 `\BlockClass{titlepage}`
 4440 `}`
 4441 `{`
 4442 `\endBlockClass`
 4443 `\LWR@printpendingfootnotes`
 4444 `}`

4445 `\end{warpHTML}`

52.3 Printing the thanks

for HTML & PRINT: 4446 `\begin{warpall}`

`\printthanks` Forces the `\thanks` to be printed.

This is necessary in a `titlingpage` environment when `\maketitle` was not used.

4447 `\newcommand*{\printthanks}{\@thanks}`

4448 `\end{warpall}`

52.4 Printing the title, etc. in HTML

The following are for printing the title, etc. in a `titlepage` or a `titlingpage` in HTML:

for HTML output: 4449 `\begin{warpHTML}`

`\printtitle`

4450 `\newcommand*{\printtitle}`

4451 `{`

4452 `\LWR@stoppars`

4453 `\LWR@htmltag{h1}%`

4454 `\@title%`

4455 `\LWR@htmltag{/h1}`

4456 `\LWR@startpars`

4457 `}`

`\LWR@printthetitle` A private version which prints the title without footnotes, used to title each HTML page.

4458 `\newcommand*{\LWR@printthetitle}`

4459 `{`

4460 `\LWR@stoppars`

4461 `\LWR@htmltag{h1}%`

4462 `\thetitle%`

4463 `\LWR@htmltag{/h1}`

4464 `\LWR@startpars`

4465 `}`

`\printauthor` HTML version.

4466 `\newcommand*{\printauthor}{`

The entire author block is contained in a <div> named author:

```
4467 \begin{BlockClass}{author}
```

\and finishes one author and starts the next:

```
4468 \renewcommand{\and}{%
4469 \end{BlockClass}
4470 \begin{BlockClass}{oneauthor}
4471 }
```

Individual authors are contained in a <div> named oneauthor:

```
4472 \begin{BlockClass}{oneauthor}
4473 \@author
4474 \end{BlockClass}
4475 \end{BlockClass}
4476 }
```

\printdate

```
4477 \newcommand*\printdate{%
4478 \begin{BlockClass}{titledate}
4479 \@date
4480 \end{BlockClass}
4481 }
```

```
4482 \end{warpHTML}
```

52.5 Printing the title, etc. in print form

The following are for printing the title, etc. in a titlepage or a titlingpage in print form:

for PRINT output: 4483 \begin{warpprint}

\printtitle

```
4484 \newcommand*\printtitle{{\Huge\@title}}
```

\printauthor Print mode.

```
4485 \newcommand*\printauthor{
4486   {\large\begin{tabular}[t]{c}\@author\end{tabular}}}
```

`\printdate`

```
4487 \newcommand*{\printdate}{\small\textit{\@date}}
4488 \end{warpprint}
```

52.6 `\maketitle` for HTML output

An HTML `<div>` of class `titlepage` is used.

`\thanks` are a form of footnotes used in the title page. See section 45 for other kinds of footnotes.

See `\thanksmarkseries{series}`, below, to set the style of the footnote marks.

for HTML output: 4489 `\begin{warppHTML}`

```
4490 \if@titlepage
4491 \newcommand{\LWR@setfootnoteseries}{%
4492   \renewcommand\thefootnote{\@arabic\c@footnote}%
4493 }
4494 \else
4495 \newcommand{\LWR@setfootnoteseries}{%
4496   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
4497 }
4498 \fi
```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
4499 \newcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
4500 \LWR@setfootnoteseries%
4501 \def\@makefnmark{\textsuperscript{\thefootnote}}
```

`\thefootnote` \Rightarrow `\nameuse{arabic}{footnote}`, or
`\thefootnote` \Rightarrow `\nameuse{fnsymbol}{footnote}`

Redefine the footnote text:

```
4502 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
4503 \textsuperscript{\@thefnmark} \LWR@origspace{1in}
```

```
\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)
```

Print the text:

```
4504 ##1%
4505 }%
4506 }
```

```
\@fnsymbol {<counter>}
```

Re-defined to use an HTML entity for the double vertical bar symbol. The original definition used `\|` which was not being seen by `pdftotext`.

```
4507 \def\@fnsymbol#1{\ifcase#1\or *\or \HTMLentity{dagger}\or \HTMLentity{Dagger}\or
4508 \HTMLentity{sect}\or \HTMLentity{para}\or \text{\HTMLUnicode{2016}}\or
4509 **\or \HTMLentity{dagger}\HTMLentity{dagger} \or
4510 \HTMLentity{Dagger}\HTMLentity{Dagger} \else\@ctrerrr\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.

```
4511 \renewcommand*\maketitle}{%
```

An HTML titlepage `<div>` is used for all classes.

```
4512 \begin{titlepage}
```

Set up special patches:

```
4513 \LWR@maketitlesetup
```

Typeset the title, etc:

```
4514 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
4515 \@thanks
```

Close the HTML titlepage div and cleanup:

```

4516 \end{titlepage}
4517 \setcounter{footnote}{0}%
4518 \global\let\thanks\relax
4519 \global\let\maketitle\relax
4520 \global\let\@maketitle\relax
4521 \global\let\@thanks\@empty
4522 \global\let\@author\@empty
4523 \global\let\@date\@empty
4524 \global\let\@title\@empty
4525 \global\let\title\relax
4526 \global\let\author\relax
4527 \global\let\date\relax
4528 \global\let\and\relax
4529 }

```

`\@maketitle` HTML mode. Typesets the title, etc.:

```

4530 \DeclareDocumentCommand{\@maketitle}{-}{%
4531   \LWR@stoppars\LWR@htmltag{h1}
4532   \@title
4533   \LWR@htmltag{/h1}\LWR@startpars
4534   \begin{BlockClass}{author}
4535     \renewcommand{\and}{
4536       \end{BlockClass}
4537       \begin{BlockClass}{oneauthor}
4538     }
4539     \begin{BlockClass}{oneauthor}
4540       \@author
4541     \end{BlockClass}
4542   \end{BlockClass}
4543   \begin{BlockClass}{titledate}
4544     \@date
4545   \end{BlockClass}
4546 }

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```

4547 \newcommand*{\LWR@titlingmaketitle}{%

```

Keep pending footnotes out of the title block:

```

4548 \@thanks

```

Set up special patches:

```

4549 \LWR@maketitlesetup

```

Typeset the title, etc:

```
4550 \@maketitle
```

Immediately generate any \thanks footnotes:

```
4551 \@thanks
4552 }
```

```
4553 \end{warpHTML}
```

52.7 \published and \subtitle

`\subtitle` and `\published` To add `\subtitle` and `\published` to the titlepage, load the titling package and use `\AddSubtitlePublished` in the preamble.

The default `lwarp.css` has definitions for the `published` and `subtitle` classes.

After titling is loaded, `\AddSubtitlePublished` is created, which when used then creates a number of additional macros, and also assigns some of the titling hooks. 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`.

△ titling hooks

for HTML & PRINT: 4554 `\begin{warpall}`

`\AddSubtitlePublished` Adds `\published` and `\subtitle`, and related.

```
4555 \AfterPackage{titling}{
4556 \newcommand*{\AddSubtitlePublished}{%
4557
4558 \newcommand{\@published}{}
4559
4560 \newcommand{\published}[1]{\gdef\@published{##1}}
4561
4562 \renewcommand*{\maketitlehooka}{\printpublished}
4563
4564 \newcommand*{\printpublished}{%
4565 \warpprintonly{\begin{center}\unskip}%
4566 \begin{BlockClass}{published}%
4567 \warpprintonly{\large\itshape}%
4568 \@published%
4569 \end{BlockClass}%
4570 \warpprintonly{\end{center}}}%
4571 }
```

```

4572
4573 \newcommand{\@subtitle}{}
4574
4575 \newcommand{\subtitle}[1]{\gdef\@subtitle{##1}}
4576
4577 \renewcommand*\maketitlehookb{\printsubtitle}
4578
4579 \newcommand*\printsubtitle{%
4580 \warpprintonly{\begin{center}\unskip}%
4581 \begin{BlockClass}{subtitle}%
4582 \warpprintonly{\Large\itshape}%
4583 \@subtitle%
4584 \end{BlockClass}%
4585 \warpprintonly{\end{center}}%
4586 }
4587
4588 }% \AddSubtitlePublished
4589 }% AfterPackage

4590 \end{warpall}

```

53 Abstract

The following code replaces the \TeX default, and will itself be replaced later if the abstract package is loaded.

for HTML output: 4591 \begin{warpHTML}

`\abstractname` User-redefinable title for the abstract.

Also over-written by the babel package.

```
4592 \providecommand*\abstractname{Abstract}
```

Env `abstract`

```

4593 \DeclareDocumentEnvironment{abstract}{}
4594 {
4595 \LWR@forcenewpage
4596 \BlockClass{abstract}
4597 \BlockClassSingle{abstracttitle}{\abstractname}
4598 }
4599 {
4600 \endBlockClass
4601 }

```

```
4602 \end{warpHTML}
```

54 Quote and verse

54.1 Citations and attributions

`\attribution` for use inside quote, quotation, verse:

```
ex: \attribution{author name} --- \citetitle{book name}
```

for HTML output:

```
4603 \begin{warpHTML}
4604 \newcommand{\attribution}[1]{%
4605 \InlineClass{attribution}{--\,#1}}% emdash
4606 \end{warpHTML}
```

for PRINT output:

```
4607 \begin{warpprint}
4608 \newcommand{\attribution}[1]{\textsc{---\,#1}}
4609 \end{warpprint}
```

`\citetitle` for use inside quote, quotation, verse:

for HTML output:

```
4610 \begin{warpHTML}
4611 \newcommand{\citetitle}[1]{%
4612 \InlineClass{citetitle}{--\,#1}}% emdash
4613 \end{warpHTML}
```

for PRINT output:

```
4614 \begin{warpprint}
4615 \newcommand{\citetitle}[1]{\textsl{---\,#1}}
4616 \end{warpprint}
```

54.2 Quotes, quotations

for HTML output:

```
4617 \begin{warpHTML}
```

Env `quote`

```
4618 \renewenvironment*{quote}
4619 {
4620 \LWR@forcenewpage
4621 \LWR@htmlblocktag{blockquote}
4622 }
4623 {\LWR@htmlblocktag{/blockquote}}
```

```

4624
4625 \renewenvironment*{quotation}
4626 {
4627 \LWR@forcenewpage
4628 \LWR@htmlblocktag{blockquotation}
4629 }
4630 {\LWR@htmlblocktag{/blockquotation}}

4631 \end{warpHTML}

```

54.3 Verse

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

```

\begin{warpHTML}

\let\attrib\attribution

\end{warpHTML}

```

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

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

for HTML & PRINT: 4632 `\begin{warpall}`

The following lengths may be set in either print or HTML output, but are only used in HTML. This allows the user to set `\vleftskip` and `\leftmargini` for print output, and optionally select different values for HTML.

<p>Len <code>\TMLvleftskip</code></p>	<p>Sets <code>\vleftskip</code> inside a <code>verse</code> environment in HTML.</p>
---------------------------------------	--

```
4633 \newlength{\HTMLvleftskip}
4634 \setlength{\HTMLvleftskip}{1em}
```

Len `\TMLleftmargini` Sets `\leftmargini` inside a verse environment in HTML.

```
4635 \newlength{\HTMLleftmargini}
4636 \setlength{\HTMLleftmargini}{4.5em}
```

```
4637 \end{warpall}
```

55 Verbatim

for HTML & PRINT: 4638 `\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.

```
4639 \newlength{\VerbatimHTMLWidth}
4640 \setlength{\VerbatimHTMLWidth}{4in}
4641 \end{warpall}
```

for HTML output: 4642 `\begin{warpHTML}`

Bool `LWR@verbtags` Used to temporarily turn off verbatim tags while doing `\verbatiminput` in the HTML head.

```
4643 \newbool{LWR@verbtags}
4644 \booltrue{LWR@verbtags}
```

`\LWR@atbeginverbatim` [*style*] {*class*}

Encloses a verbatim environment with the given CSS class.

```
4645 \newcommand*{\LWR@atbeginverbatim}[2] []
4646 {%
```

Avoid excessive space between lines:

```
4647 \setlength{\parskip}{0ex}%
```

Stop generating HTML paragraph tags:

```
4648 \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.

```
4649 \ifbool{LWR@verbtags}{%
4650   \LWR@htmltag{pre class="#2"
4651   \ifthenelse{\equal{#1}{}}{style="#1"}{}}%
4652 }% pre
4653 }{}
```

Use a mono-spaced font to preserve horizontal positioning. If horizontal alignment is important for the user, use a mono-spaced font in the CSS for the `verse` class.

Also turn off `babel-french` extra space before punctuation:

```
4654 \begingroup%
4655 \LWR@origttfamily%
4656 \LWR@FBcancel%
```

Do not produce HTML tags for `\hspace` inside a `verse` par. Restore plain \TeX `\hspace` functionality:

```
4657 \LetLtxMacro{\hspace}{\LWR@orighspace}%
4658 }
```

`\LWR@afterendverbatim` Finishes enclosing a `verbatim` environment.

```
4659 \newcommand*{\LWR@afterendverbatim}{%
```

Remove excess vertical space at the end of the `pre`:

```
4660 \endgroup%
4661 \unskip%
```

At the end of the environment, close the `pre`:

```
4662 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/pre}
4663
4664 }{}
```

Resume regular paragraph handling:

```
4665 \LWR@startpars%
4666 }
```

`\verbatiminput` $\langle filename \rangle$

Patch `\verbatiminput` to add HTML tags:

```

4667 \let\LWRV@origverbatim@input\verbatim@input
4668
4669 \renewcommand{\verbatim@input}[2]{%
4670 \ifbool{LWR@verbtags}{\LWR@forcenewpage}{}%
4671 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4672 \LWRV@origverbatim@input{#1}{#2}%
4673 \unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim%
4674 }

```

Env `verbatim`

```

4675 \AfterEndPreamble{
4676 \LWR@traceinfo{Patching verbatim.}
4677 \AtBeginEnvironment{verbatim}{%
4678 \LWR@forcenewpage
4679 \LWR@atbeginverbatim{verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
4680 }
4681 \AfterEndEnvironment{verbatim}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
4682 }

4683 \end{warpHTML}

```

56 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: 4684 `\begin{warpHTML}`

`\@begintheorem` $\langle name \rangle$ $\langle number \rangle$

```

4685 \renewcommand{\@begintheorem}[2]{%
4686 \LWR@forcenewpage
4687 \BlockClass{theoremcontents}
4688 \InlineClass{theoremlabel}{#1\ #2\ }
4689 }

```

```

\@opargbegintheorem  {\langle name\rangle} {\langle number\rangle} {\langle oparg\rangle}

4690 \renewcommand{\@opargbegintheorem}[3]{%
4691 \LWR@forcenewpage
4692 \BlockClass{theoremcontents}
4693 \InlineClass{theoremlabel}{#1\ #2\ (#3)\ }
4694 }

\@endtheorem

4695 \renewcommand*\@endtheorem}{%
4696 \endBlockClass% theoremcontents
4697 }

4698 \end{warpHTML}

```

57 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}

```

57.1 Itemize

for HTML output: 4699 `\begin{warpHTML}`

`\LWR@itemizeitem` [*⟨label⟩*]

Handles `\item` inside an itemize or enumerate.

See `\LWR@openparagraph` where extra `\hspace` is used to leave room for the label while inside a list during paragraph construction.

```
4700 \newcommand*\LWR@itemizeitem}{%
4701 \LWR@stoppars%
4702 \LWR@startnewdepth{\LWR@depthlistitem}{\LWR@printcloseitemize{}}%
4703 \LWR@htmltag{li}%
4704 \LWR@startpars%
4705 \LWR@origitem%
4706 }
```

Env `itemize` [*⟨options⟩*]

```
4707 \newcommand*\LWR@itemizestart}{%
4708 \LWR@stoppars%
4709 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
4710 \LWR@htmltag{ul style="list-style-type:none"}%
4711 \LWR@startpars%
4712 \let\item\LWR@itemizeitem%
4713 }
4714
4715 \newcommand*\LWR@itemizeend}{%
4716 \LWR@stoppars%
4717 \LWR@closeprevious{\LWR@depthlistitem}%
4718 \LWR@closeoneprevious{}%
4719 \LWR@startpars%
4720 }
```

57.2 Enumerate

An HTML unordered list is used with customized \TeX -generated labels.

Env `enumerate` [*⟨options⟩*]

```
4721 \newcommand*\LWR@enumeratestart}{%
4722 \LWR@stoppars%
4723 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printcloseitemize{}}%
```

```

4724 \LWR@htmltag{ul style="list-style-type:none"}{}}%
4725 \LWR@startpars%
4726 \let\item\LWR@itemizeitem%
4727 }
4728
4729 \newcommand*{\LWR@enumerateend}{-%
4730 \LWR@stoppars%
4731 \LWR@closeprevious{\LWR@depthlistitem}%
4732 \LWR@closeoneprevious{}%
4733 \LWR@startpars%
4734 }

```

57.3 Description

`\LWR@descitem` [*(label)*] Handles an `\item` inside a description.

```

4735 \newcommand*{\LWR@descitem}[1][ ]{-%
4736 {-%
4737 \LWR@stoppars%
4738 \LWR@setlatestname{#1}%
4739 \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.

```

4740 \LetLtxMacro{\hspace}{\LWR@nohspace}%

```

Process the original `\item` code:

```

4741 \LWR@origitem[]%

```

Restore `\hspace` for use in the item text:

```

4742 \LetLtxMacro{\hspace}{\LWR@hspace}%

```

Be sure the label doesn't print to the left of the rest of the file:

```

4743 \LWR@orighspace{1in}
4744 \LWR@htmltag{dt}#1\LWR@htmltag{/dt}%
4745 \LWR@orignewline%
4746 \LWR@htmltag{dd}%
4747 \LWR@startpars%
4748 }

```

Env `description` [*(options)*]

```

4749 \newcommand*{\LWR@descriptionstart}{%
4750 \LWR@stoppars%
4751 \LWR@pushoneclose{\LWR@depthlist}{\LWR@printclosedescription{}}%
4752 \LWR@htmltag{dl}%
4753 \LWR@startpars%
4754 \let\item\LWR@descitem%
4755 }
4756
4757 \newcommand*{\LWR@descriptionend}{%
4758 \LWR@stoppars%
4759 \LWR@closeprevious{\LWR@depthlistitem}%
4760 \LWR@closeoneprevious{}}%
4761 \LWR@startpars%
4762 }

```

`\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`.

```

4763 \newcommand*{\LWR@patchlists}{%
4764 \LetLtxMacro{\LWR@origitem}{\item}
4765 \AtBeginEnvironment{itemize}{\LWR@itemizestart}
4766 \AtEndEnvironment{itemize}{\LWR@itemizeend}
4767 \AtBeginEnvironment{enumerate}{\LWR@enumeratestart}
4768 \AtEndEnvironment{enumerate}{\LWR@enumerateend}
4769 \AtBeginEnvironment{description}{\LWR@descriptionstart}
4770 \AtEndEnvironment{description}{\LWR@descriptionend}
4771 }
4772 \end{warpHTML}

```

58 Tabular

This is arguably the most complicated part of the entire package. Numerous tricks are employed to handle the syntax which is involved.

58.1 Limitations

Tabular mostly works as expected, but pay special attention to the following, especially if working with environments, macros inside tabulars, `multirows`, `* column`

specifiers, siunitx S columns, or the packages multirow, longtable, supertabular, or xtab.

Defining environments:

⚠ misplaced alignment
alignment tab character &

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

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

⚠ tabular inside another
environment

- When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing environment ends, resulting in corrupted HTML rows. To fix this, use \ResumeTabular as follows. This is ignored in print mode.

```
\StartDefiningTabulars % because & is used in a
definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\
}
{
\TabularMacro\ResumeTabular
left & right \\
\endtabular
}
\EndDefiningTabulars
```

Cell contents:

⚠ paragraphs

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

⚠ \multirow

- For multirow, insert \mrowcell into any empty multi-row cells. This will be a null function for the print output, and is a placeholder for parsing the table for HTML output.
- The multirow documentation regarding colored cells recommends using a negative number of rows. This will not work with lwarp, so \warpprintonly and \warpHTMLonly must be used to make versions for print and HTML.

⚠ \multicolumn &
\multirow

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

```
\multicolumnrow{2}{c}[c]{3}[0]{1in}[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. Note that recent versions of `multirow` include a new optional `vposn` argument. See section 174.2.

⚠ [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 types](#)

⚠ * [column specification](#)

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

Rules:

[\multirow with rules](#)

[rule at last row](#)

[\cmidrule width, trim](#)

[\cmidrule and](#)
[\multicolumn](#)

⚠ [\warpprintonly](#)
[misplaced \noalign](#)

- If a `multirow` reaches to the bottom of a table, and `\bottomrule` does not go over to that edge, try adding a line of empty cells below the `\bottomrule`. (This may be a browser bug.)
- If a `\midrule` is desired after the last row, an additional row of blank cells must be used.
- `\cmidrule` does not support width or trim options due to CSS limitations.

- `\cmidrule` borders are generated by the individual cells on the following row, and so do not necessarily work correctly when the following row has `\multicolumn` cells below `\cmidrule` borders.

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

```
This & That \endhead
```

to

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

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

Other:

[longtable headings](#)

- `tabularx` ignores the width, but X columns do produce paragraph columns or multicolumns.
- For `longtable`, place headings and footings which do not apply to HTML inside `\warpprintonly{}`.

⚠ **S columns**

- For S columns (from the siunitx package), while producing print output, anything non-numeric must be placed inside {} braces, including commands such as \multirow. While producing HTML output, though, anything placed inside braces is not seen by lwarp's tabular handling algorithm. To resolve this problem, make a copy of the row, with one version for print output, containing the extra braces, and another version for HTML output, without the extra braces, such as:

```
\warpprintonly{1 & 2 & {\multirow{2}{2cm}{Text}} & 3 \\}
\warpHTMLonly{1 & 2 & \multirow{2}{2cm}{Text} & 3 \\}
```

58.2 Token lookahead

Used by \LWR@futurenonpacelet to look at the next token.

for HTML output: 4773 \begin{warpHTML}

\LWR@mynexttoken

```
4774 \newcommand\LWR@mynexttoken\relax
```

\futurelet copies the next token then executes a function to analyze

\LWR@futurenonpacelet does the same, but ignores intervening white space

Based on the booktabs style:

\LWR@futurenonpacelet

```
4775 \def\LWR@futurenonpacelet#1{\def\LWR@cs{#1}%
4776 \afterassignment\LWR@fnslone\let\nexttoken= }
4777 \def\LWR@fnslone{\expandafter\futurelet\LWR@cs\LWR@fnsltwo}
4778 \def\LWR@fnsltwo{%
4779 \expandafter\ifx\LWR@cs\@sptoken\let\next=\LWR@fnslthree%
4780 \else\let\next=\nexttoken\fi\next}
4781 \def\LWR@fnslthree{\afterassignment\LWR@fnslone\let\next= }
```

\LWR@getmynexttoken Looks ahead and copies the next token into \LWR@mynexttoken.

```
4782 \newcommand*{\LWR@getmynexttoken}{%
4783 \LWR@traceinfo{\LWR@getmynexttoken}%
4784 % nothing must follow this next line
4785 \LWR@futurenonpacelet\LWR@mynexttoken\LWR@tabledatacolumnntag
4786 }
```

58.3 Booleans

Bool LWR@startedrow True if should print a row tag before this column.

```
4787 \newbool{LWR@startedrow}
4788 \boolfalse{LWR@startedrow}
```

Bool LWR@doinghline True if the next row will have an hline above it.

```
4789 \newbool{LWR@doinghline}
4790 \boolfalse{LWR@doinghline}
```

Bool LWR@doingtbrule True if the next row will have a top/bottom rule above it.

```
4791 \newbool{LWR@doingtbrule}
4792 \boolfalse{LWR@doingtbrule}
```

Bool LWR@tableparcell True if are handling a paragraph inside a table cell, so must close the paragraph tag before moving on.

```
4793 \newbool{LWR@tableparcell}
```

Bool LWR@skippingmrowcell True if are doing an empty multi-row cell, and thus there is no data tag to close.

```
4794 \newbool{LWR@skippingmrowcell}
```

Bool LWR@skipatbang True if just finished a \multicolumn so should not print the trailing @ or ! columns.

```
4795 \newbool{LWR@skipatbang}
```

Bool LWR@intabularmetadata True if are in a tabular but not in a data cell. Used to prevent extra HTML breaks if not inside table data.

```
4796 \newbool{LWR@intabularmetadata}
4797 \boolfalse{LWR@intabularmetadata}
```

58.4 Handling &, @, and !

For technical discussion regarding problems redefining \&, See:

<http://tex.stackexchange.com/questions/11638/>

[where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860](http://tex.stackexchange.com/questions/11638/where-do-i-find-futurelets-nasty-behaviour-documented/11860#11860)

\LWR@closetabledatacell If LWR@skippingmrowcell then there is no data tag to close. Otherwise, close any paragraphs, then close the data tag.

```

4798 \newcommand*{\LWR@closetabledatacell}{%
4799 \global\booltrue{\LWR@intabularmetadata}%
4800 \ifbool{\LWR@exitingtabular}{}%
4801 {% not exiting tabular
4802   \ifbool{\LWR@skippingmrowcell}{}%
4803   {% not skippingmrowcell

```

Insert any < then any @ and ! column contents:

```

4804   \unskip%
4805   \LWR@getexparray{\LWR@colafterspec}{\theLWR@tablecolspos}%
4806 % % \LWR@getexparray{\LWR@colatspec}{\theLWR@tablecolspos}%
4807 % \LWR@printatbang{at}{\theLWR@tablecolspos}%
4808 % % \LWR@getexparray{\LWR@colbangspec}{\theLWR@tablecolspos}%
4809 % \LWR@printatbang{bang}{\theLWR@tablecolspos}%

```

Close paragraphs:

```

4810   \ifbool{\LWR@tableparcell}{\LWR@stoppars}{}%
4811   \global\boolfalse{\LWR@tableparcell}%

```

Close the table data cell. Skip the @ and ! cells if are closing a multicolumn cell.

```

4812   \leavevmode\unskip\LWR@htmltag{/td}\LWR@orignewline%
4813   \ifbool{\LWR@skipatbang}%
4814   {\boolfalse{\LWR@skipatbang}}%
4815   {%
4816     \LWR@printatbang{at}{\theLWR@tablecolspos}%
4817     \LWR@printatbang{bang}{\theLWR@tablecolspos}%
4818   }% not skipping at or bang
4819   }% not skipping mrowcell
4820 }% not exiting tabular
4821 \global\boolfalse{\LWR@skippingmrowcell}%
4822 }

```

LWR@tabulardepth tracks whether & is being used inside a tabular.

```

4823 \newcounter{\LWR@tabulardepth}
4824 \setcounter{\LWR@tabulardepth}{0}
4825

```

When not used inside a tabular, & performs its original function as recorded here (with catcode 4).

```

4826 \let\LWR@origampmacro&

```

```

4827 \end{warpHTML}

```

58.4.1 Localizing & catcodes

for HTML & PRINT: 4828 `\begin{warpall}`

 **misplaced alignment
tab character &**

Place `\StartDefiningTabulars` and `\EndDefiningTabulars` before and after defining macros or environments which include the tabular & character in their definitions.

The catcode of & must be changed before the definitions begin, and must be restored afterwards. Doing so avoids the error

```
misplaced alignment tab character &
```

`\StartDefiningTabulars` Place before defining something with & in it.

```
4829 \newcommand{\StartDefiningTabulars}{%
4830 \LWR@traceinfo{StartDefiningTabulars}
4831 \warpHTMLonly{\catcode'\&=\active}%
4832 }
```

`\EndDefiningTabulars` Place after defining something with & in it.

```
4833 \newcommand{\EndDefiningTabulars}{%
4834 \LWR@traceinfo{EndDefiningTabulars}
4835 \warpHTMLonly{\catcode'\&=4}%
4836 }
```

```
4837 \end{warpall}
```

58.4.2 Handling &

for HTML output: 4838 `\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.

```
4839 \newcommand*{\LWR@tabularampersand}{%
4840 \LWR@traceinfo{\LWR@tabularampersand}%
4841 \ifthenelse{\cnttest{\value{\LWR@tabulardepth}}{>}{0}}{
4842 {%
```

If not skipping a multirow cell, close the current data cell.

```
4843 \unskip%
4844 \LWR@closetabledatacell%
```

Move to the next column.

```
4845 \addtocounter{LWR@tablecolspos}{1}%
```

Look at the next token to decide multi or single column data tag.

```
4846 \LWR@getmynexttoken%
4847 }%
```

If not inside a tabular, performs the original action:

```
4848 {\LWR@origampmacro}%
4849 }
```

& is left with its original catcode for now.

tikz package seems to require & be left alone until after tikz has been loaded. Also, cleveref uses the ampersand in one of its options.

& is made active inside a tabular.

& is left alone when in math alignments.

58.5 Handling \\

Inside tabular, \\ is redefined to \LWR@tabularendofline

Throws away options \\[dim] or *

```
\LWR@tabularendofline
```

```
4850 \NewDocumentCommand{\LWR@tabularendofline}{s o}
4851 {%
4852 \LWR@closetabledatacell%
```

Finish the previous row:

```
4853 \LWR@htmltag{/tr}\LWR@orignewline
4854 \global\booltrue{LWR@intabularmetadata}
```

Not yet started a table row:

```
4855 \global\boolfalse{LWR@startedrow}
```

Additional setup:

```

4856 \global\boolfalse{LWR@doinghline}%
4857 \global\boolfalse{LWR@doingtbrule}%
4858 \LWR@clearmidrules%

```

Start at first column:

```
4859 \setcounter{LWR@tablecolspos}{1}
```

Look at the next token to decide between single column data tag or a special case:

```

4860 \LWR@getmynexttoken%
4861 }

```

58.6 Variables

```

4862 \newcommand*{\LWR@colsresult}{}%temp storage for column format results
4863 \newcommand*{\LWR@pposition}{}
4864 \newcommand*{\LWR@pleft}{}
4865 \newcommand*{\LWR@pright}{}

```

`\LWR@tablecolspec` Holds the parsed column specification, of total width `LWR@tabletotalcols`, not counting @ and ! columns.

Will contain a string such as `llrrccpc`, exactly one letter per \LaTeX table column, without @, !, >, <, or the vertical pipe.

This is indexed by the counter `LWR@tabletotalcols`.

```
4866 \newcommand*{\LWR@tablecolspec}{}

```

`\LWR@strresult` Holds the result of `Str` functions.

```

4867 \newcommand*{\LWR@strresult}{}
4868 \newcommand*{\LWR@strresulttwo}{}

```

`\LWR@origcolspec` Holds the original column specs given to `tabular`.

```
4869 \newcommand*{\LWR@origcolspec}{}

```

`Ctr` `LWR@tablecolwidth` Holds the width of the table columns specification.

This is the number of tokens, including one for each @ etc. column, and also one each for the parameters of p, @, etc. columns, and three for each D column.

(This is not the total # of \LaTeX columns in the table.)

```
4870 \newcounter{LWR@tablecolwidth}
```

Ctrl LWR@tablecolspos Where are currently looking into the table column specification. Index starts at 1.

```
4871 \newcounter{LWR@tablecolspos}
```

Ctrl LWR@tabletotalcols Holds the final number of table columns, not counting @ and ! columns.

```
4872 \newcounter{LWR@tabletotalcols}
```

Ctrl LWR@tabletotalcolsnext Holds the next column while parsing. Is one more than LWR@tabletotalcols.

```
4873 \newcounter{LWR@tabletotalcolsnext}
```

LWR@colatspec A data array of specifications for @ columns. The leftmost's index is leftedge, the others are counter values. See section 32.

LWR@colbangspec A data array of specifications for ! columns. The leftmost's index is leftedge, the others are counter values. See section 32.

LWR@colbeforespec A data array of specifications for > columns.

LWR@colafterspec A data array of specifications for < columns.

58.7 Parsing @, >, <, ! columns

Holds the parsed argument for @, >, <, or ! columns:

```
4874 \newcommand*{\LWR@colparameter}{}
```

\LWR@parseatcolumn Handles @{text} columns.

```
4875 \newcommand*{\LWR@parseatcolumn}{%
```

Move to the next token after the '@':

```
4876 \LWR@traceinfo{at column}%
```

```
4877 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
4878 \LWR@traceinfo{about to read the next token:}%
```

```
4879 \expandarg%
```

```
4880 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4881 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4882 \LWR@traceinfo{have now read the next token}%
4883 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
4884 {% left edge of the table:
4885   \LWR@traceinfo{at the left edge}%
4886   \LWR@setexparray{LWR@colatspec}{leftedge}{\LWR@colparameter}%
4887   \LWR@traceinfo{at the left edge: %
4888   \LWR@getexparray{LWR@colatspec}{leftedge}}%
4889 }%
4890 {% not at the left edge:
4891   \LWR@traceinfo{not at the left edge}%
4892   \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4893   \LWR@traceinfo{at \theLWR@tabletotalcols: %
4894   \LWR@getexparray{LWR@colatspec}{\theLWR@tabletotalcols}}%
4895 }%
4896 \let\LWR@colparameter\relax%
4897 \booltrue{LWR@validtablecol}%
4898 }
```

\LWR@parsebangcolumn Handles !{text} columns.

```
4899 \newcommand*{\LWR@parsebangcolumn}{%
```

Move to the next token after the '!':

```
4900 \LWR@traceinfo{bang column}%
4901 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token into \LWR@strresult, expanding once:

```
4902 \LWR@traceinfo{about to read the next token:}%
4903 \expandarg%
4904 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]
4905 \fullexpandarg%
```

Store the result into a data array, expanding once out of \LWR@strresult:

```
4906 \LWR@traceinfo{have now read the next token}%
4907 \ifthenelse{\cnttest{\value{LWR@tabletotalcols}}=0}
4908 {% left edge of the table:
4909   \LWR@traceinfo{at the left edge}%
4910   \LWR@setexparray{LWR@colbangspec}{leftedge}{\LWR@colparameter}%
4911 }%
4912 {% not at the left edge:
```

```

4913 \LWR@traceinfo{not at the left edge}%
4914 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcols}{\LWR@colparameter}%
4915 \LWR@traceinfo{bang \theLWR@tabletotalcols: \LWR@colparameter!}%
4916 }%
4917 \let\LWR@colparameter\relax%
4918 \booltrue{LWR@validtablecol}%
4919 }

```

`\LWR@parsebeforecolumn` Handles `>{text}` columns.

```
4920 \newcommand*{\LWR@parsebeforecolumn}{%
```

Move to the next token after the '>':

```
4921 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```

4922 \expandarg%
4923 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4924 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```

4925 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{\LWR@colparameter}%
4926 \let\LWR@colparameter\relax%
4927 \booltrue{LWR@validtablecol}%
4928 }

```

`\LWR@parseaftercolumn` Handles `<{text}` columns.

```
4929 \newcommand*{\LWR@parseaftercolumn}{%
```

Move to the next token after the '<':

```
4930 \addtocounter{LWR@tablecolspos}{1}%
```

Read the next token, expanding once into `\LWR@strresult`:

```

4931 % \StrChar{#1}{\theLWR@tablecolspos}[\LWR@colparameter]
4932 \expandarg%
4933 \StrChar{\LWR@origcolspec}{\theLWR@tablecolspos}[\LWR@colparameter]%
4934 \fullexpandarg%

```

Store the result into a data array, expanding once out of `\LWR@strresult`:

```

4935 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcols}{LWR@colparameter}%
4936 \let\LWR@colparameter\relax%
4937 \booltrue{LWR@validtablecol}%
4938 }

```

`\LWR@parseskipcolumn` Handles columns to skip, such as the vertical bar.

```

4939 \newcommand*{\LWR@parseskipcolumn}{%
4940 \booltrue{LWR@validtablecol}%
4941 }

```

58.8 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.

```

4942 \newcommand*{\LWR@parsenormalcolumn}[1]{%
4943 \appto\LWR@tablecolspec{#1}%
4944 \addtocounter{LWR@tabletotalcols}{1}%
4945 \addtocounter{LWR@tabletotalcolsnext}{1}%
4946 \LWR@traceinfo{normal column \theLWR@tabletotalcols: #1}%
4947 \LWR@setexparray{LWR@colatspec}{\theLWR@tabletotalcolsnext}{}%
4948 \LWR@setexparray{LWR@colbangspec}{\theLWR@tabletotalcolsnext}{}%
4949 \LWR@setexparray{LWR@colbeforespec}{\theLWR@tabletotalcolsnext}{}%
4950 \LWR@setexparray{LWR@colafterspec}{\theLWR@tabletotalcolsnext}{}%
4951 \booltrue{LWR@validtablecol}%
4952 }

```

58.9 Parsing ‘p’, ‘m’, or ‘b’ columns

`\LWR@parsepcolumn` $\langle thiscolumn \rangle$ The width will be ignored.

```

4953 \newcommand*{\LWR@parsepcolumn}[1]{%

```

Converts to the given column type:

```

4954 \LWR@parsenormalcolumn{#1}%

```

Skips the following width token:

Table 6: Tabular baseline

l	p	m	b	r
			bot	
		mid	bot	
l	par	mid	bot	r
	par	mid		
	par			

```
4955 \addtocounter{LWR@tablecolspos}{1}%
4956 }
```

58.10 Parsing ‘D’ columns

From the dcolumn package.

`\LWR@parseDcolumn` $\langle\{thiscolumn\}\rangle$ The three parameters will be ignored.

```
4957 \newcommand*{\LWR@parseDcolumn}[1]{%
```

Converts to the given column type.

```
4958 \LWR@parsenormalcolumn{#1}%
```

Skips the following three parameters.

```
4959 \addtocounter{LWR@tablecolspos}{3}%
4960 }
```

58.11 Parsing the column specifications



HTML CSS cannot exactly match the \TeX concept of a baseline for a table row. Table 6 shows the \TeX results for various vertical-alignment choices, with the baseline of the first column drawn across all the columns for comparison. See the p column specification in table 7 for details.

Table 7 describes how each kind of column is converted to HTML.

Bool `LWR@validtablecol` True if found a valid table column type.

Table 7: Tabular HTML column conversions

l, r, c:	Converted to table cells without paragraph tags. Uses CSS <code>vertical-align:middle</code> so that top or bottom-aligned cells may go above or below this cell.
p:	Converted to table cells with paragraph tags. Ref: Table 6, \LaTeX places the top line of a parbox aligned with the rest of the text line, so CSS <code>vertical-align:bottom</code> is used to have the HTML result appear with the paragraph extending below the L, R, C cells at the middle, if possible. This may be confusing as a P cell may not top-align with an L,R,C cell in the HTML conversion, especially in the presence of a B cell, and two P cells side-by-side will be aligned at the bottom instead of the top. Some adjustment of the CSS may be desired, changing <code>td.tdp</code> , <code>td.tdP</code> , <code>td.tdprule</code> , and <code>td.tdPrule</code> to <code>vertical-align: middle</code> . Another possibility is to change L,R,C, and P to <code>vertical-align: top</code> and not worry about the alignment of B and M cells or trying to approximate \LaTeX baselines.
m:	With paragraph tags, CSS <code>vertical-align:middle</code> .
b:	With paragraph tags, CSS <code>vertical-align:top</code> so that the bottom of the text is closest to the middle of the text line.
P, M, B:	Horizontally-centered versions.
S:	Converted to 'r'. From the <code>siunitx</code> package.
D:	Converted to 'c'. From the <code>dcolumn</code> package.
@, !, >, <:	One each, in that order.
Unknown:	Converted to 'l'.
\newcolumn:	Currently treated as unknown.

```
4961 \newbool{LWR@validtablecol}
```

```
\LWR@parsetablecols {<colspecs>}
```

Scans the column specification left to right.

Builds `\LWR@tablecolspec` with the final specification, one column per entry. The number of final columns is stored in `LWR@tabletotalcols`.

```
4962 \newcommand*{\LWR@parsetablecols}[1]{%
4963 \LWR@traceinfo{LWR@parsetablecols started}%
```

Remember the original supplied column spec:

```
4964 \renewcommand*{\LWR@origcolspec}{#1}%
```

Clear the parsed resulting column spec:

```
4965 \renewcommand*{\LWR@tablecolspec}{}%
```

Total number of columns found so far. Also pre-initialize the first several columns of specs:

```
4966 \setcounter{LWR@tabletotalcols}{0}%
4967 \setcounter{LWR@tabletotalcolsnext}{1}%
4968 \LWR@setexparray{LWR@colatspec}{leftedge}{}%
4969 \LWR@setexparray{LWR@colatspec}{1}{}%
4970 \LWR@setexparray{LWR@colatspec}{2}{}%
4971 \LWR@setexparray{LWR@colatspec}{3}{}%
4972 \LWR@setexparray{LWR@colbangspec}{leftedge}{}%
4973 \LWR@setexparray{LWR@colbangspec}{1}{}%
4974 \LWR@setexparray{LWR@colbangspec}{2}{}%
4975 \LWR@setexparray{LWR@colbangspec}{3}{}%
4976 \LWR@setexparray{LWR@colbeforespec}{1}{}%
4977 \LWR@setexparray{LWR@colbeforespec}{2}{}%
4978 \LWR@setexparray{LWR@colbeforespec}{3}{}%
4979 \LWR@setexparray{LWR@colafterspec}{1}{}%
4980 \LWR@setexparray{LWR@colafterspec}{2}{}%
4981 \LWR@setexparray{LWR@colafterspec}{3}{}%
```

Starting at the first column specification:

```
4982 \setcounter{LWR@tablecolspos}{1}%
```

Place the colspecs string length into `\LWR@strresult`, and remember the number of characters in the column specification:

```

4983 \LWR@traceinfo{about to StrLen}%
4984 \noexpandarg%
4985 \StrLen{#1}[\LWR@strresult]%
4986 \fullexpandarg%
4987 \LWR@traceinfo{finished StrLen}%
4988 \setcounter{LWR@tablecolwidth}{\LWR@strresult}%

```

Scan through the column specifications:

```
4989 \whiledo{\not\value{LWR@tablecolspos}>\value{LWR@tablecolwidth}}{%
```

Place the next single-character column type into \LWR@strresult:

```

4990 \noexpandarg%
4991 \StrChar{#1}{\theLWR@tablecolspos}[\LWR@strresult]%
4992 \LWR@traceinfo{position \arabic{LWR@tablecolspos}: \LWR@strresult}%
4993 \fullexpandarg%

```

Not yet found a valid column type

```
4994 \boolfalse{LWR@validtablecol}%
```



Note that the parameter for a p{spec} column is a token list which will NOT match l,c,r,p.

```

4995 \IfStrEq{\LWR@strresult}{l}{\LWR@parsenormalcolumn{l}}{}%
4996 \IfStrEq{\LWR@strresult}{c}{\LWR@parsenormalcolumn{c}}{}%
4997 \IfStrEq{\LWR@strresult}{r}{\LWR@parsenormalcolumn{r}}{}%
4998 \IfStrEq{\LWR@strresult}{L}{\LWR@parsenormalcolumn{l}}{}%
4999 \IfStrEq{\LWR@strresult}{C}{\LWR@parsenormalcolumn{c}}{}%
5000 \IfStrEq{\LWR@strresult}{R}{\LWR@parsenormalcolumn{r}}{}%
5001 \IfStrEq{\LWR@strresult}{J}{\LWR@parsenormalcolumn{l}}{}%
5002 \IfStrEq{\LWR@strresult}{S}{\LWR@parsenormalcolumn{r}}{}%
5003 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@parseatcolumn}{}%
5004 \IfStrEq{\LWR@strresult}{!}{\LWR@parsebangcolumn}{}%
5005 \IfStrEq{\LWR@strresult}{>}{\LWR@parsebeforecolumn}{}%
5006 \IfStrEq{\LWR@strresult}{<}{\LWR@parseaftercolumn}{}%
5007 \IfStrEq{\LWR@strresult}{|}{\LWR@parseskipcolumn}{}%
5008 \IfStrEq{\LWR@strresult}{p}{\LWR@parsepcolumn{p}}{}%
5009 \IfStrEq{\LWR@strresult}{m}{\LWR@parsepcolumn{m}}{}%
5010 \IfStrEq{\LWR@strresult}{b}{\LWR@parsepcolumn{b}}{}%

```

From the dcolumn package:

```
5011 \IfStrEq{\LWR@strresult}{D}{\LWR@parseDcolumn{c}}{%
```

From the tabularx package. X column has no parameter, but will be given paragraph tags.

```
5012 \IfStrEq{\LWR@strresult}{X}{\LWR@parsenormalcolumn{X}}{}%
```

Many people define centered versions “P”, “M”, and “B”:

```
\newcolumnntype{P}[1]{>\centering\arraybackslash}p{#1}}
```

```
5013 \IfStrEq{\LWR@strresult}{P}{\LWR@parsepcolumn{P}}{}%
```

```
5014 \IfStrEq{\LWR@strresult}{M}{\LWR@parsepcolumn{M}}{}%
```

```
5015 \IfStrEq{\LWR@strresult}{B}{\LWR@parsepcolumn{B}}{}%
```

If this column was an invalid column type, convert it to an l column:

```
5016 \ifbool{\LWR@validtablecol}{}%
```

```
5017   \LWR@traceinfo{invalid column type: \LWR@strresult}%
```

```
5018   \LWR@parsenormalcolumn{l}%
```

```
5019 }%
```

```
5020 \addtocounter{\LWR@tablecolspos}{1}%
```

```
5021 }%
```

```
5022 }%
```

58.12 Starting a new row

`\LWR@maybenewtablerow` If have not yet started a new table row, begin one now. Creates a new row tag, adding a class for hline or tbrule if necessary.

```
5023 \newcommand*{\LWR@maybenewtablerow}
```

```
5024 {%
```

```
5025 \ifbool{\LWR@startedrow}%
```

```
5026 {}% started the row
```

```
5027 {% not started the row
```

Remember that now have started the row:

```
5028   \global\booltrue{\LWR@startedrow}%
```

Create the row tag, with a class if necessary.

```
5029   \global\booltrue{\LWR@intabularmetadata}%
```

```
5030   \ifbool{\LWR@doinghline}%
```

```
5031   {\LWR@htmltag{tr class="hline"{}}\LWR@orignewline}%
```

```
5032   {% not doing hline
```

```
5033     \ifbool{\LWR@doingtbrule}%
```

```
5034     {\LWR@htmltag{tr class="tbrule"{}}\LWR@orignewline}%
```

```
5035     {\LWR@htmltag{tr}\LWR@orignewline}%
```

```
5036   }% end of not doing hline
```

```
5037 }% end of not started the row
5038 }
```

58.13 Printing at or bang tags

`\LWR@printatbang` $\langle at -or- bang \rangle$ $\langle index \rangle$

```
5039 \newcommand*\LWR@printatbang}[2]{%
5040 \edef\LWR@atbangspec{\LWR@getexparray{\LWR@col#1spec}{#2}}
5041 \LWR@traceinfo{atbang: !\LWR@atbangspec!}
5042 \ifdefempty{\LWR@atbangspec}%
5043 % \ifthenelse{\isempty{\LWR@atbangspec}}%
5044 {}%
5045 {%
5046   \LWR@htmltag{td class="td#1%
5047   \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%
5048   "%}
5049   \LWR@atbangspec%
5050   \LWR@htmltag{/td}\LWR@orignewline%
5051 }%
5052 }%
```

58.14 Data opening tag

`\LWR@tabledatasinglecolumnntag` Print a table data opening tag with style for alignment

```
5053 \newcommand*\LWR@tabledatasinglecolumnntag{%
5054 {%
5055 \LWR@maybe newtablerow%
```

Don't start a new paragraph tag if have already started one:

```
5056 \ifbool{\LWR@intabularmetadata}%
5057 {%
```

If have found the end of tabular command, do not create the next data cell:

```
5058   \ifbool{\LWR@exitingtabular}{}%
5059   {% not exiting tabular
```

Print the @ and ! contents before first column:

```
5060       \ifthenelse{\cnttest{\value{\LWR@tablecolspos}}=1}{%
```

```

5061      {%
5062          \LWR@printatbang{at}{leftedge}%
5063          \LWR@printatbang{bang}{leftedge}%
5064      }% left edge
5065      {}% not left edge

```

Fetch the current column's alignment character into \LWR@strresult:

```

5066      \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}[\LWR@strresult]%

```

print the start of a new table data cell:

```

5067      \LWR@htmltag{td class="td%

```

append this column's spec:

```

5068      \LWR@strresult%

```

If this column has a cmidrule, add “rule” to the end of the HTML class tag:

```

5069      \ifthenelse{\equal{\LWR@getexpparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}%
5070      {rule}%
5071      {}%
5072      "{}}%

```

If this is a p, m, b, or X column, allow paragraphs:

```

5073      \ifthenelse{%
5074          \equal{\LWR@strresult}{p}\OR%
5075          \equal{\LWR@strresult}{m}\OR%
5076          \equal{\LWR@strresult}{b}\OR%
5077          \equal{\LWR@strresult}{P}\OR%
5078          \equal{\LWR@strresult}{M}\OR%
5079          \equal{\LWR@strresult}{B}\OR%
5080          \equal{\LWR@strresult}{X}%
5081      }%
5082      {% allow pars
5083          \LWR@startpars%
5084          \global\booltrue{\LWR@tableparcell}%
5085      }% allow pars
5086      {}% no pars

```

Print the > contents:

```

5087      \LWR@getexpparray{\LWR@colbeforespec}{\theLWR@tablecolspos}%
5088      \global\boolfalse{\LWR@intabularmetadata}%
5089      }% not exiting tabular
5090      {}% in tabular metadata
5091      }%

```

58.15 Midrules

`LWR@midrules` `LWR@midrules` is a data array (section 32) of columns containing Y if a midrule should be created for each column.

Ctrl `LWR@midrulecounter` Indexes across the `LWR@midrules` data array.

```
5092 \newcounter{LWR@midrulecounter}
```

`\LWR@clearmidrules` Start new midrules. Called at beginning of tabular and also at `\`.

Clears all `LWR@midrules` markers for this line.

```
5093 \newcommand*{\LWR@clearmidrules}
5094 {%
5095 \setcounter{LWR@midrulecounter}{1}%
5096 \whiledo{%
5097 \cnttest{\value{LWR@midrulecounter}}{<=}{\value{LWR@tablecolwidth}}%
5098 }%
5099 {%
5100 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{}%
5101 \addtocounter{LWR@midrulecounter}{1}%
5102 }%
5103 }
```

`\LWR@subcmidrule` `{<width>}{<trim>}{<leftcolumn>}{<rightcolumn>}`

Marks `LWR@midrules` data array elements to be “Y” from left to right columns.

```
5104 \newcommand*{\LWR@subcmidrule}[4]{%
5105 \setcounter{LWR@midrulecounter}{#3}%
5106 \whiledo{\cnttest{\value{LWR@midrulecounter}}{<=}{#4}}%
5107 {%
5108 \LWR@setexparray{LWR@midrules}{\theLWR@midrulecounter}{Y}%
5109 \addtocounter{LWR@midrulecounter}{1}%
5110 }% end of the whiledo
5111 }
```

`\LWR@docmidrule` `[<width>]{<trim>}{<leftcolumn-rightcolumn>}`

Marks `LWR@midrules` array elements to be “Y” from left to right columns.

```
5112 \NewDocumentCommand{\LWR@docmidrule}{o d() >{\SplitArgument{1}{-}}m}%
5113 {\LWR@subcmidrule{#1}{#2}{#3}}
```

58.16 Multicolumns

58.16.1 Parsing multicolumns

```
5114 \newcounter{LWR@tablemulticolwidth}
```

Indexes into the multicolumn specification:

```
5115 \newcounter{LWR@tablemulticolspos}
```

`\LWR@printmcoltype` `{<colspec>}` Print any valid column type found. Does not print @, !, >, or < columns or their associated tokens.

This is printed as part of the table data tag's class.

```
5116 \newcommand*{\LWR@printmcoltype}[1]{%
5117 \LWR@traceinfo{lwr@printmcoltype -#1-}%
```

Get one token of the column spec:

```
5118 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Add to the HTML tag depending on which column type is found:

```
5119 \IfStrEq{\LWR@strresult}{l}{l}{}%
5120 \IfStrEq{\LWR@strresult}{c}{c}{}%
5121 \IfStrEq{\LWR@strresult}{r}{r}{}%
5122 \IfStrEq{\LWR@strresult}{p}{p}{}%
5123 \IfStrEq{\LWR@strresult}{m}{m}{}%
5124 \IfStrEq{\LWR@strresult}{b}{b}{}%
5125 \IfStrEq{\LWR@strresult}{P}{P}{}%
5126 \IfStrEq{\LWR@strresult}{M}{M}{}%
5127 \IfStrEq{\LWR@strresult}{B}{B}{}%
5128 \IfStrEq{\LWR@strresult}{S}{r}{}%
5129 \IfStrEq{\LWR@strresult}{X}{p}{}%
5130 \LWR@traceinfo{lwr@printmcoltype done}%
5131 }
```

`\LWR@multicolpartext` Print the data with paragraph tags:

```
5132 \newcommand*{\LWR@multicolpartext}{%
5133 \LWR@startpars%
5134 \LWR@multicoltext%
5135 \LWR@stoppars%
5136 }
```

`\LWR@multicolother` $\langle colspec \rangle$ For @, !, >, <, print the next token without paragraph tags:

```
5137 \newcommand*\LWR@multicolother}[1]{%
5138 \addtocounter{LWR@tablemulticolspos}{1}%
5139 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
5140 \LWR@strresult%
```

A valid column data type was found:

```
5141 \booltrue{LWR@validtablecol}%
5142 }
```

`\LWR@multicolskip` Nothing to print for this column type.

```
5143 \newcommand*\LWR@multicolskip}{%
```

A valid column data type was found:

```
5144 \booltrue{LWR@validtablecol}%
5145 }
```

`\LWR@printmccoldata` $\langle colspec \rangle$ Print the data for any valid column type found.

```
5146 \newcommand*\LWR@printmccoldata}[1]{%
5147 \LWR@traceinfo{lw@printmccoldata -#1}%
```

Not yet found a valid column type:

```
5148 \boolfalse{LWR@validtablecol}%
```

Get one token of the column spec:

```
5149 \StrChar{#1}{\theLWR@tablemulticolspos}[\LWR@strresult]%
```

Print the text depending on which column type is found. Also handles @, >, < as it comes to them.

```
5150 \IfStrEq{\LWR@strresult}{l}{\LWR@multicoltext}{}%
5151 \IfStrEq{\LWR@strresult}{c}{\LWR@multicoltext}{}%
5152 \IfStrEq{\LWR@strresult}{r}{\LWR@multicoltext}{}%
5153 \IfStrEq{\LWR@strresult}{D}{}%
5154 \addtocounter{LWR@tablemulticolspos}{3}% skip parameters
5155 \LWR@multicoltext%
5156 }{}%
5157 \IfStrEq{\LWR@strresult}{p}{\LWR@multicolparttext}{}%
5158 \IfStrEq{\LWR@strresult}{m}{\LWR@multicolparttext}{}%
5159 \IfStrEq{\LWR@strresult}{b}{\LWR@multicolparttext}{}%
```

```

5160 \IfStrEq{\LWR@strresult}{P}{\LWR@multicolparttext}{}%
5161 \IfStrEq{\LWR@strresult}{M}{\LWR@multicolparttext}{}%
5162 \IfStrEq{\LWR@strresult}{B}{\LWR@multicolparttext}{}%
5163 \IfStrEq{\LWR@strresult}{S}{\LWR@multicolparttext}{}%
5164 \IfStrEq{\LWR@strresult}{X}{\LWR@multicolparttext}{}%
5165 \IfStrEq{\LWR@strresult}{|}{\LWR@multicolskip}{}%
5166 \IfStrEq{\LWR@strresult}{\detokenize{@}}{\LWR@multicolother{#1}}{}%
5167 \IfStrEq{\LWR@strresult}{\detokenize{!}}{\LWR@multicolother{#1}}{}%
5168 \IfStrEq{\LWR@strresult}{\detokenize{>}}{\LWR@multicolother{#1}}{}%
5169 \IfStrEq{\LWR@strresult}{\detokenize{<}}{\LWR@multicolother{#1}}{}%

```

If an invalid column type:

```
5170 \ifbool{LWR@validtablecol}{\LWR@multicoltext}{%
```

Tracing:

```

5171 \LWR@traceinfo{lw@printmccoldata done}%
5172 }

```

`\parsemulticolumnalignment` $\{ \langle 1: \text{colspec} \rangle \} \{ \langle 2: \text{printresults} \rangle \}$

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.

```

5173 \newcommand*{\LWR@parsemulticolumnalignment}[2]{%
5174 \setcounter{LWR@tablemulticolspos}{1}%
5175 \StrLen{#1}[\LWR@strresult]%
5176 \setcounter{LWR@tablemulticolwidth}{\LWR@strresult}%

```

Scan across the tokens in the column spec:

```

5177 \whiledo{%
5178 \not\value{LWR@tablemulticolspos}>\value{LWR@tablemulticolwidth}%
5179 }%
5180 {%

```

Execute the assigned print function for each token in the column spec:

```
5181 #2{#1}%
```

Move to the next token in the column spec:

```

5182 \addtocounter{LWR@tablemulticolspos}{1}%
5183 }%
5184 }

```

58.16.2 Multicolumn factored code

```
5185 \newcommand{\LWR@multicoltext}{}

```

```
\LWR@domulticolumn [{1: vpos}] [{2: #rows}] [{3: numcols}] [{4: colspec}] [{5: text}]

```

```
5186 \NewDocumentCommand{\LWR@domulticolumn}{o o m m +m}{%
5187 \LWR@traceinfo{lwr@domulticolumn -#1- -#2- -#3- -#4-}%

```

Remember the text to be inserted, and remember that a valid column type was found:

```
5188 \renewcommand{\LWR@multicoltext}{%
5189 #5%
5190 \booltrue{LWR@validtablecol}%
5191 }%

```

Row processing:

```
5192 \LWR@maybenewtablerow%

```

Begin the opening table data tag:

```
5193 \LWR@htmltag{td colspan="#3" %

```

```
5194 \IfValueT{#2}{ % rows?
5195 rowspan="#2" %

```

```
5196 \IfValueT{#1}{% vpos?
5197 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
5198 \ifstrequal{#1}{t}{style="vertical-align:top" }{}%
5199 }% vpos?
5200 }% rows?

```

```
5201 class="td%

```

Print the column type:

```
5202 \LWR@parsemulticolumnalignment{#4}{\LWR@printmccoltype}%

```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag.

If this position had a “Y” then add “rule”.

```
5203 \ifthenelse{\equal{\LWR@getexparray{LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{}%

```

Close the class tag’s opening quote:

```

5204 "%
5205 }% end of the opening table data tag
5206 \global\boolfalse{LWR@intabularmetadata}%
5207 \LWR@parsemulticolumnalignment{#4}{\LWR@printmccoldata}%
5208 }

```

58.16.3 Multicolumn

```
\LWR@htmlmulticolumn {<numcols>} {<alignment>} {<text>}
```

```

5209 \NewDocumentCommand{\LWR@htmlmulticolumn}{m m +m}%
5210 {%

```

Figure out how many extra HTML columns to add for @ and ! columns:

```
5211 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn tag:

```
5212 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{#2}{#3}%
```

Move to the next \LaTeX column:

```

5213 \addtocounter{LWR@tablecolspos}{#1}%
5214 \addtocounter{LWR@tablecolspos}{-1}%

```

Skip any trailing @ or ! columns for this cell:

```

5215 \booltrue{LWR@skipatbang}%
5216 }

```

58.16.4 Longtable captions

longtable captions use `\multicolumn`.

Bool `LWR@starredlongtable` Per the caption package, step the counter if `longtable*`.

```

5217 \newbool{LWR@starredlongtable}
5218 \boolfalse{LWR@starredlongtable}

```

Per the caption package. User-redefinable float type.

```
5219 \providecommand*\LTcapttype{table}
```

`\LWR@longtabledatacaptiontag` * [*toc entry*] {*caption*}

```
5220 \NewDocumentCommand{\LWR@longtabledatacaptiontag}{s o +m}
5221 {%
```

Remember the latest name for `\nameref`:

```
5222 \IfValueTF{#2}{% optional given?
5223   \ifthenelse{\equal{#2}{}}{% optional empty?
5224     {\LWR@setlatestname{#3}}% empty
5225     {\LWR@setlatestname{#2}}% given and non-empty
5226 }% optional given
5227 {\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:

```
5228 \LWR@tabularhtmlcolumns{1}{\theLWR@tabletotalcols}
```

Create the multicolumn tag:

```
5229 \LWR@domulticolumn{\theLWR@tabhtmlcoltotal}{P}{% \LWR@domulticolumn
5230 \IfBooleanTF{#1}% star?
```

Star version, show a caption but do not make a LOT entry:

```
5231 {% yes star
5232   \LWR@htmlblocktag{figcaption}%
5233   #3%
5234   \LWR@htmlblocktag{/figcaption}%
5235 }%
5236 {% No star:
```

Not the star version:

Don't step the counter if `\caption[]` {A caption.}

```
5237   \ifbool{LWR@starredlongtable}%
5238   {%
5239   \ifthenelse{\equal{#2}{}}% TOC entry
5240   {}%
5241   {%
5242     \refstepcounter{LTcaption}%
5243     \protected@edef\@currentlabel{%
5244       \csuse{p@LTcaption}\csuse{the\LTcaption}}%
5245     }%
5246   }{}%
```

Create an HTML caption. Afterwards, maybe make a LOT entry.

```
5247 \LWR@htmlblocktag{figcaption}%
5248 \csuse{fnum@\LTcaption}\CaptionSeparator#3%
5249 \LWR@htmlblocktag{/figcaption}%
```

See if an optional caption was given:

```
5250 \ifthenelse{\equal{#2}{}}% TOC entry empty
```

if the optional caption was given, but empty, do not form a TOC entry

```
5251 {}%
```

If the optional caption was given, but might only be []:

```
5252 {% TOC entry not empty
5253 \IfNoValueTF{#2}% No TOC entry?
```

The optional caption is []:

```
5254 {% No TOC entry
5255 \addcontentsline%
5256 {\csuse{ext@\LTcaption}}%
5257 {\LTcaption}%
5258 {%
5259 \protect\numberline%
5260 {\csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5261 {\ignorespaces #3\protect\relax}%
5262 }%
5263 }% end of No TOC entry
```

The optional caption has text enclosed:

```
5264 {% yes TOC entry
5265 \addcontentsline%
5266 {\csuse{ext@\LTcaption}}%
5267 {\LTcaption}%
5268 {%
5269 \protect\numberline%
5270 {\csuse{p@\LTcaption}\csuse{the\LTcaption}}%
5271 {\ignorespaces #2\protect\relax}%
5272 }%
5273 }% end of yes TOC entry
5274 }% end of TOC entry not empty
5275 }% end of no star
```

Skip any trailing @ or ! columns for this cell:

```

5276 \booltrue{LWR@skipatbang}%
5277 }% end of \LWR@domulticolumn
5278
5279 \addtocounter{LWR@tablecolspos}{\theLWR@tabletotalcols}
5280 \addtocounter{LWR@tablecolspos}{-1}
5281
5282 }

```

58.16.5 Counting HTML tabular columns

The \LaTeX specification for a table includes a number of columns separated by the `&` character. These columns differ in content from line to line. Additional virtual columns may be specified by the special `@` and `!` columns. These columns are identical from line to line, but may be skipped during a multicolumn cell.

For HTML output, `@` and `!` columns are placed into their own tabular columns. Thus, a \LaTeX `\multicolumn` command may span several additional `@` and `!` columns in HTML output. These additional columns must be added to the total number of columns spanned by an HTML multi-column data cell.

```

5283 \newcounter{LWR@tabhtmlcolindex}
5284 \newcounter{LWR@tabhtmlcolend}
5285 \newcounter{LWR@tabhtmlcoltotal}

```

```
\LWR@tabularhtmlcolumns {<starting  $\LaTeX$  column>} {<number  $\LaTeX$  columns>}
```

Compute the total number of HTML columns being spanned, considering the starting \LaTeX table column and the number of \LaTeX tabular columns being spanned. Any `@` and `!` columns within this span are included in the total count. The resulting number of HTML columns is returned in the counter `LWR@tabhtmlcoltotal`.

```
5286 \newcommand*{\LWR@tabularhtmlcolumns}[2]{%
```

Count the starting index, compute ending index, and begin with the count being the \LaTeX span, to which additional `@` and `!` columns may be added:

```

5287 \setcounter{LWR@tabhtmlcolindex}{#1}%
5288 \setcounter{LWR@tabhtmlcoltotal}{#2}%
5289 \setcounter{LWR@tabhtmlcolend}{#1}%
5290 \addtocounter{LWR@tabhtmlcolend}{#2}%

```

Walk across the \LaTeX columns looking for `@` and `!` columns:

```
5291 \whiledo{\value{LWR@tabhtmlcolindex}<\value{LWR@tabhtmlcolend}}{%
```

Temporarily define a macro equal to the @ specification for this column:

```
5292 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{\theLWR@tabhtmlcolindex}}%
```

If the @ specification is not empty, add to the count:

```
5293 \ifdefempty{\LWR@atbangspec}{-}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Likewise for the ! columns:

```
5294 \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{\theLWR@tabhtmlcolindex}}%
5295 \ifdefempty{\LWR@atbangspec}{-}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
```

Move to the next \TeX column:

```
5296 \addtocounter{LWR@tabhtmlcolindex}{1}%
5297 }%
```

If at the left-most column, also skip the leftmost @ and ! cells:

```
5298 \ifthenelse{\value{LWR@tablecolspos}=1}{%
5299   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colatspec}{leftedge}}%
5300   \ifdefempty{\LWR@atbangspec}{-}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5301   \edef\LWR@atbangspec{\LWR@getexparray{LWR@colbangspec}{leftedge}}%
5302   \ifdefempty{\LWR@atbangspec}{-}{\addtocounter{LWR@tabhtmlcoltotal}{1}}%
5303 }-}{%
5304 }
```

```
5305 \end{warpHTML}
```

58.17 Multicolumnrow

A print-mode version is defined here, and is also used during HTML output while inside a `lateximage`.

See section 174 for the HTML versions.

for HTML & PRINT: 5306 `\begin{warpall}`

```
\multicolumnrow {<1:cols>} {<2:halign>} [<3:vpos>] {<4:numrows>} [<5:bigstruts>] {<6:width>} [<7:fixup>]
{<8:text>}
```

For discussion of the use of `\DeclareExpandableDocumentCommand`, see:
<https://tex.stackexchange.com/questions/168434/problem-with-abbreviation-of-multirow-and-multicolumn-latex>

After the user may have

```

5307 \AtBeginDocument{

\@ifundefined{@xmultirow} determines if multirow was never loaded.

5308 \@ifundefined{@xmultirow}
5309 {}% no version of multirow was loaded
5310 {% \@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.

5311 \@ifpackageloaded{multirow}{% v2.0 or newer
5312 \@ifpackagelater{multirow}{2016/09/01}% 2016/09/27 for v2.0
5313 {% v2.0+:
5314 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5315   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5316 {\multicolumn{#1}{#2}{\@xmultirow[#3][#4][#5][#6][#7][#8]}}%
5317 }
5318 {% loaded but older, probably not executed:
5319 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5320   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5321 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5][#6][#7][#8]}}%
5322 }
5323 }% 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.

5324 {% v1.6 or older did not \ProvidePackage
5325 \DeclareExpandableDocumentCommand{\LWR@origmulticolumnrow}%
5326   {+m +m +0{c} +m +0{0} +m +0{Opt} +m}%
5327 {\multicolumn{#1}{#2}{\@xmultirow{#4}[#5][#6][#7][#8]}}%
5328 }

The user-level interface. This is provided if the HTML version was not already given.

5329 \providecommand*{\multicolumnrow}{\LWR@origmulticolumnrow}
5330 }% \@xmultirow defined, so multirow was loaded
5331 }% AtBeginDocument

5332 \end{warpall}

```

58.18 Utility macros inside a table

for HTML output: 5333 `\begin{warpHTML}`

```
5334 \newcommand*\LWR@donothing{}
5335 \newcommand*\LWR@domidrule{\booltrue{LWR@doinghline}}
5336 \newcommand*\LWR@dotbrule{\booltrue{LWR@doingtbrule}}
```

In case bigdelim is not loaded:

```
5337 \newcommand*\ldelim{}
5338 \newcommand*\rdelim{}

5339 \end{warpHTML}
```

58.19 Special-case tabular markers

for HTML & PRINT: 5340 `\begin{warpall}`

`\TabularMacro` Place this just before inserting a custom macro in a table data cell. Doing so tells lwarp not to automatically start a new HTML table data cell yet. See section 7.7.

```
5341 \let\TabularMacro\relax

5342 \end{warpall}
```

`\ResumeTabular` Used to resume tabular entries after resuming an environment.

 **tabular inside another environment** When creating a new environment which contains a tabular environment, lwarp's emulation of the tabular does not automatically resume when the containing environment ends, resulting in corrupted HTML rows. To fix this, use `\ResumeTabular` as follows. This is ignored in print mode.

```

\StartDefiningTabulars % because & is used in a definition
\newenvironment{outerenvironment}
{
\tabular{cc}
left & right \\\
}
{
\TabularMacro\ResumeTabular
left & right \\\
\endtabular
}
\EndDefiningTabulars

```

for HTML output: 5343 \begin{warpHTML}

```

5344 \newcommand*\ResumeTabular{%
5345 \global\boolfalse{LWR@exitingtabular}%
5346 \LWR@getmynexttoken%
5347 }

```

5348 \end{warpHTML}

for PRINT output: 5349 \begin{warpprint}

```

5350 \newcommand*\ResumeTabular{}

```

5351 \end{warpprint}

58.20 Checking for a new table cell

for HTML output: 5352 \begin{warpHTML}

\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:

```

5353 \newbool{LWR@exitingtabular}
5354
5355 \newcommand*\LWR@tabledatacolumnntag%
5356 {%
5357 \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:

```
5358 \let\mynext\LWR@tabledatasinglecolumnntag%
```

If exiting the tabular:

```
5359 \ifdefequal{\LWR@mynexttoken}{\end}%
5360   {\booltrue{LWR@exitingtabular}}{}%
```

longtable can have a caption in a cell

```
5361 \ifdefequal{\LWR@mynexttoken}{\caption}%
5362   {\let\mynext\LWR@donothing}{}%
```

Look for other things which would not start a table cell:

```
5363 \ifdefequal{\LWR@mynexttoken}{\multicolumn}%
5364   {\let\mynext\LWR@donothing}{}%
5365 \ifdefequal{\LWR@mynexttoken}{\multirow}%
5366   {\let\mynext\LWR@donothing}{}%
5367 \ifdefequal{\LWR@mynexttoken}{\multicolumnrow}%
5368   {\let\mynext\LWR@donothing}{}%
```

if come to an `\mrowcell`, this is a cell to be skipped over

```
5369 \ifdefequal{\LWR@mynexttoken}{\mrowcell}%
5370   {\let\mynext\LWR@donothing}{}%
5371 %
5372 \ifdefequal{\LWR@mynexttoken}{\TabularMacro}%
5373   {\let\mynext\LWR@donothing}{}%
5374 %
5375 \ifdefequal{\LWR@mynexttoken}{\hline}%
5376   {\let\mynext\LWR@donothing}{}%
5377 %
5378 \ifdefequal{\LWR@mynexttoken}{\toprule}%
5379   {\let\mynext\LWR@donothing}{}%
5380 %
5381 \ifdefequal{\LWR@mynexttoken}{\midrule}%
5382   {\let\mynext\LWR@donothing}{}%
5383 %
5384 \ifdefequal{\LWR@mynexttoken}{\cmidrule}%
5385   {\let\mynext\LWR@donothing}{}%
5386 %
5387 \ifdefequal{\LWR@mynexttoken}{\specialrule}%
5388   {\let\mynext\LWR@donothing}{}%
5389 %
5390 \ifdefequal{\LWR@mynexttoken}{\cline}%
5391   {\let\mynext\LWR@donothing}{}%
5392 %
```

```

5393 \ifdefequal{\LWR@mynexttoken}{\bottomrule}%
5394     {\let\LWR@mynext\LWR@donothing}{}%
5395 %
5396 \ifdefequal{\LWR@mynexttoken}{\warpprintonly}%
5397     {\let\LWR@mynext\LWR@donothing}{}%
5398 %
5399 \ifdefequal{\LWR@mynexttoken}{\warpHTMLonly}%
5400     {\let\LWR@mynext\LWR@donothing}{}%
5401 %
5402 \ifdefequal{\LWR@mynexttoken}{\ldelim}%
5403     {\let\LWR@mynext\LWR@donothing}{}%
5404 %
5405 \ifdefequal{\LWR@mynexttoken}{\rdelim}%
5406     {\let\LWR@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:

```

5407 \mynext%
5408 }

5409 \end{warpHTML}

```

58.21 `\mrowcell`

`\mrowcell`  The user must insert `\mrowcell` into any multirow cells which must be skipped. This command has no action during print output.

for HTML & PRINT:

```

5410 \begin{warpall}
5411 \newcommand*{\mrowcell}{}
5412 \end{warpall}

```

58.22 New `\tabular` definition

for HTML output:

```

5413 \begin{warpHTML}

```

These are default definitions in case `booktabs` is not loaded, and are not expected to used, but must exist as placeholders.

```

5414 \newcommand*{\LWR@origtoprule}[1]{\hline}

```

```

5415 \newcommand*\LWR@origmidrule}[1]{\hline}
5416 \LetLtxMacro\LWR@origcmidrule\cline
5417 \newcommand*\LWR@origbottomrule}[1]{\hline}
5418 \newcommand*\LWR@origaddlinespace}[1][]{\hline}
5419 \newcommand*\LWR@origmorecmidrules{}
5420 \newcommand*\LWR@origspecialrule}[3]{\hline}

```

`\LWR@hline` The definition of `\hline` depends on whether `tabls` has been loaded. If so, optional space below the line may be specified, but will be ignored.

```

5421 \AtBeginDocument{
5422 \@ifpackageloaded{lwarp-tbls}
5423 {\newcommand*\LWR@hline}[1][]{\LWR@domidrule\LWR@getmynexttoken}
5424 {\newcommand*\LWR@hline}{\LWR@domidrule\LWR@getmynexttoken}
5425 }% AtBeginDocument

```

Env `LWR@tabular` [*⟨vertical⟩*] {*⟨colspecs⟩*}

The new `tabular` environment will be `\let` in `\LWR@LwarpStart`, since `siunitx` might redefine `tabular` in the user's document.

```

5426 \StartDefiningTabulars
5427
5428 \newenvironment*\LWR@tabular}[2]{}
5429 {}
5430 \LWR@traceinfo{\LWR@tabular started}%
5431 \addtocounter{\LWR@tabulardepth}{1}%

```

Not yet started a table row:

```
5432 \global\boolfalse{\LWR@startedrow}%
```

Not yet doing an hline:

```
5433 \global\boolfalse{\LWR@doinghline}%
```

Not yet doing a top/bottom rule:

```
5434 \global\boolfalse{\LWR@doingtbrule}%
```

For `babel-french`:

```
5435 \LWR@FBcancel%
```

Have not yet found the end of `tabular` command:

```
5436 \boolfalse{\LWR@exitingtabular}%
```

Create the table tag:

```
5437 \global\booltrue{LWR@intabularmetadata}%
5438 \LWR@forcenewpage
5439 \LWR@htmlblocktag{table}%
```

Parse the table columns:

```
5440 \LWR@parsetablecols{#2}%
```

Table col spec is: \LWR@tablecolspec which is a string of 11ccrr, etc.

Do not place the table inside a paragraph:

```
5441 \LWR@stoppars%
```

Track column # for setting text-align:

```
5442 \setcounter{LWR@tablecolspos}{1}%
```

Start looking for midrules:

```
5443 \LWR@clearmidrules%
```

\\ becomes a macro to end the table row:

```
5444 \LetLtxMacro{\\}{\LWR@tabularendofline}%
```

The following may appear before a data cell is created, so after doing their actions, we look ahead with \LWR@getmynexttoken to see if the next token might create a new data cell:

The optional parameter for \hline supports the tabs package.

```
5445 \LWR@traceinfo{LWR@tabular: redefining macros}%
5446 \renewcommand*{\hline}{\LWR@hline}%
5447 %
5448 \RenewDocumentCommand{\cline}{m}%
5449 {\LWR@docmidrule{##1}\LWR@getmynexttoken}%
5450
5451 \DeclareDocumentCommand{\toprule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
5452
5453 \DeclareDocumentCommand{\midrule}{o}{\LWR@domidrule\LWR@getmynexttoken}%
5454
5455 \DeclareDocumentCommand{\cmidrule}{o d() m}%
5456 {\LWR@docmidrule{##1}{##2}{##3}\LWR@getmynexttoken}%
5457
5458 \DeclareDocumentCommand{\bottomrule}{o}{\LWR@dotbrule\LWR@getmynexttoken}%
```

```

5459 \DeclareDocumentCommand{\addlinespace}{o}{-}
5460 \DeclareDocumentCommand{\morecmidrules}{-}{-}
5461 \DeclareDocumentCommand{\specialrule}{m m m}{\LWR@domidrule\LWR@getmynexttoken}

```

The following create data cells and will have no more data in this cell, so we do not want to look ahead for a possible data cell, so do not want to use `\LWR@getmynexttoken`.

```

5462 \renewcommand{\multicolumn}{\LWR@htmlmulticolumn}%
5463 \renewcommand*{\mrowcell}{\global\booltrue{\LWR@skippingmrowcell}}%
5464 \LetLtxMacro{\caption}{\LWR@longtabledatacaptiontag}%

```

Reset for new processing:

```

5465 \global\boolfalse{\LWR@tableparcell}%
5466 \global\boolfalse{\LWR@skippingmrowcell}%
5467 \global\boolfalse{\LWR@skipatbang}%

```

Set `&` for its special meaning inside the tabular:

```

5468 \StartDefiningTabulars%
5469 \protected\gdef&{\LWR@tabularampersand}%

```

Look ahead for a possible table data cell:

```

5470 \LWR@traceinfo{\LWR@tabular: about to \LWR@getmynexttoken}%
5471 \LWR@getmynexttoken%
5472 }%

```

Ending the environment:

```

5473 {%
5474 \LWR@traceinfo{\LWR@tabular ending}%
5475 \LWR@closetabledatacell%
5476 \LWR@htmlblocktag{/tr}%
5477 \LWR@htmlblocktag{/table}%
5478 \global\boolfalse{\LWR@intabularmetadata}%

```

Unnest one level of tabular:

```

5479 \addtocounter{\LWR@tabulardepth}{-1}%

```

Restore `&` to its usual meaning:

```

5480 \protected\gdef&{\LWR@origampmacro}%
5481 \EndDefiningTabulars%
5482 \LWR@traceinfo{\LWR@tabular finished ending}%
5483 }
5484
5485 \EndDefiningTabulars

```

```
5486 \end{warpHTML}
```

58.23 Array

Pkg array

array is also automatically loaded by siunitx.

59 Cross-references

Sectioning commands have been emulated from scratch, so the cross-referencing commands are custom-written for them. Emulating both avoids several layers of patches.

The zref package is used to remember section name, file, and lateximage depth and number for each label.

Table 8 shows the data structures related to cross-referencing.

for HTML output:

```
5487 \begin{warpHTML}
```

59.1 Setup

`\@currentlabelname` To remember the most recently defined section name, description, or caption, for `\nameref`.

```
5488 \newcommand*\@currentlabelname{}
```

`\LWR@stripperperiod` `{\text}` [`<.>`]

Removes a trailing period.

```
5489 \def\LWR@stripperperiod#1.\ltx@empty#2\@nil{#1}%
```

`\LWR@setlatestname` `{\object name}`

Removes `\label`, strips any final period, and remembers the result.

```
5490 \newcommand*\LWR@setlatestname[1]{%
```

Table 8: 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 59.3), plus <code>\splabel</code> data (section 59.2):	
<code>zLWR@name</code>: The section name for this label.	
<code>zLWR@htmlfilenumber</code>: The 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 59.4.	
<code>\ref</code> and <code>\nameref</code>: Adds HTML tags. See section 59.4.	
Added by amsmath:	(print and HTML)
<code>\label</code>: Execution is delayed until the math environment is completed.	
<code>\ltx@label</code>: \LaTeX <code>\label</code> , (HTML: patched by <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 ctrs>]{\p<ctr>\the<ctr>}</code> Also see section 45.4 for use with footnotes.	
<code>\label</code>: Writes to the .aux file: <code>\newlabel{<label>@cref}{\cref@currentlabel}{\thepage}}</code>	
<code>\newlabel</code>: (Unchanged.) When the .aux file is read, sets <code>\r<label>@cref</code> .	
<code>\r<label>@cref</code>: Set to: <code>{\cref@currentlabel}{\thepage}}</code>	
Utility functions: See <code>\cref@getlabel</code> , <code>\cref@gettype</code> , <code>\cref@getcounter</code> , <code>\cref@getprefix</code> .	
Cross-referencing names: <code>\crefname</code> and <code>\Crefname</code> assign human-readable names for references to this counter type.	
Additionally patched by lwarp:	(HTML only)
<code>\cref</code>, etc.: Modified for <code>lwarp</code> . See section 69.	
<code>\label</code> inside math: See section 63.4.1.	
Footnotes: See <code>\noteentry</code> in section 45.4.	

Remove `\label` and other commands from the name, the strip any final period. See `zref-titleref` and `getttitlestring`.

```
5491 \GetTitleStringExpand{#1}%
5492 \edef\@currentlabelname{\detokenize\expandafter{\GetTitleStringResult}}%
5493 \edef\@currentlabelname{%
5494 \expandafter\LWR@stripperperiod\@currentlabelname%
5495 \ltx@empty.\ltx@empty\@nil%
5496 }%
5497 }
```

59.2 Zref setup

See:

<http://tex.stackexchange.com/questions/57194/extract-section-number-from-equation-reference>

Create a new property list called `special`:

```
5498 \zref@newlist{special}
```

Define a new property which has the name of the most recently declared section:

```
5499 \zref@newprop{zLWR@name}{\@currentlabelname}
```

Define a new property which has either a filename or a file number:

```
5500 \zref@newprop{zLWR@htmlfilenumber}{%
5501 \ifbool{FileSectionNames}{\LWR@thisfilename}{\theLWR@htmlfilenumber}%
5502 }%
```

Additional properties for `lateximages`:

```
5503 \zref@newprop{zLWR@lateximagedepth}{\arabic{LWR@lateximagedepth}}
5504 \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`:

```
5505 \zref@addprop{special}{zLWR@name}
5506 \zref@addprop{special}{zLWR@htmlfilenumber}
5507 \zref@addprop{special}{zLWR@lateximagedepth}
5508 \zref@addprop{special}{zLWR@lateximagenumber}
```

Returns the selected field:

```
5509 \newcommand*\LWR@spref}[2]{%
5510 \zref@extractdefault{#1}{#2}{??}}
```

`\LWR@nameref` $\langle label \rangle$ Returns the section name for this label:

```
5511 \newcommand*\LWR@nameref}[1]{%
5512 \LWR@spref{#1}{zLWR@name}%
5513 }
```

`\LWR@htmlfileref` $\langle label \rangle$ Returns the file number for this label:

```
5514 \newcommand*\LWR@htmlfileref}[1]{%
5515 % DO NOT USE \LWR@traceinfo HERE! Will be expanded.
5516 \LWR@spref{#1}{zLWR@htmlfilenumber}%
5517 }
```

`\LWR@lateximagedepthref` $\langle label \rangle$ Returns the lateximagedepth for this label:

```
5518 \newcommand*\LWR@lateximagedepthref}[1]{%
5519 \LWR@spref{#1}{zLWR@lateximagedepth}%
5520 }
```

`\LWR@lateximagenumberref` $\langle label \rangle$ Returns the lateximagenumber for this label:

```
5521 \newcommand*\LWR@lateximagenumberref}[1]{%
5522 \LWR@spref{#1}{zLWR@lateximagenumber}%
5523 }
```

`\LWR@splabel` $\langle label \rangle$ Sanitize the name and then creates the label:

```
5524 \newcommand*\LWR@splabel}[1]{%
5525 \LWR@setlatestname{\@currentlabelname}%
5526 \zref@labelbylist{#1}{special}}
```

59.3 Labels

`\LWR@subsublabel` $\langle label \rangle$ Creates an HTML id tag.

```
5527 \newcommand*\LWR@subsublabel}[1]{%
```

Create an HTML id tag unless are inside a lateximage, since it would appear in the image:

```
5528 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
5529 {}%
5530 {% not lateximage
```

If not doing a lateximage, create an HTML ID tag: (To be factored...)

```
5531 \ifbool{LWR@doingstartpars}%
5532 {% pars allowed
5533 \ifbool{LWR@doingapar}
5534 {% par started
5535 \LWR@htmltag{a id="#1"}\LWR@htmltag{/a}%
5536 }% par started
5537 {% par not started
5538 \LWR@stoppars%
5539 \LWR@htmltag{a id="#1"}\LWR@htmltag{/a}%
5540 \LWR@startpars%
5541 }% par not started
5542 }% pars allowed
5543 {% pars not allowed
5544 \LWR@htmltag{a id="#1"}\LWR@htmltag{/a}%
5545 }% pars not allowed
5546 }% not lateximage
5547 }
```

`\LWR@newlabel` $\langle label \rangle$ [$\langle type \rangle$]

`\label` during HTML output when not in math mode, removing extra spaces around the label, as done by regular \LaTeX `\label`.

`cleveref` later encases this to add its own cross-referencing.

The optional $\langle type \rangle$ is per the `ntheorem` package, and is ignored.

```
5548 \NewDocumentCommand{\LWR@newlabel}{m o}{%
5549 \LWR@traceinfo{LWR@newlabel: starting}%
5550 \LWR@traceinfo{LWR@newlabel: !#1!}%
5551 % \@bsphack%
```

Create a traditional \LaTeX label, as modified by `cleveref`:

```
5552 \LWR@origlabel{#1}%
```

Create a special label which holds the section number, `LWR@htmlfilenumber`, `LWR@lateximagedepth`, and `LWR@lateximagenumber`:

```

5553 \LWR@traceinfo{LWR@newlabel: filesectionnames is \ifbool{FileSectionNames}{true}{false}}%
5554 \LWR@traceinfo{LWR@newlabel: LWR@thisfilename is !\LWR@thisfilename!}%
5555 \LWR@traceinfo{LWR@newlabel: LWR@htmlfilenumber is \theLWR@htmlfilenumber}%
5556 \LWR@splabel{#1}%
5557 \LWR@subsublabel{#1}%
5558 % \@esphack%
5559 \LWR@traceinfo{LWR@newlabel: done}%
5560 }

```

59.4 References

`\LWR@startref` $\{\langle label \rangle\}$ (Common code for `\ref` and `\nameref`.)

Open an HTML tag reference to a filename, # character, and a label.

```

5561 \newcommand*{\LWR@startref}[1]
5562 {%
5563 \edef\LWR@lidref{\LWR@lateximagedepthref{#1}}%
5564 \LWR@traceinfo{LWR@startref A: !#1!}%

```

Create the filename part of the link:

```

5565 \LWR@htmltag{a href="%
5566 \LWR@traceinfo{LWR@startref B}%
5567 \LWR@htmlrefsectionfilename{#1}%
5568 \LWR@traceinfo{LWR@startref C}%
5569 \#%

```

Create the destination id:

See if `LWR@lateximagedepth` is unknown:

```

5570 \LWR@traceinfo{LWR@startref D: !#1!}%
5571 \ifthenelse{\equal{\LWR@lidref}{??}}%

```

“??” if `LWR@lateximagedepth` is unknown, so create a link with an unknown destination:

```

5572 {%
5573 \LWR@traceinfo{LWR@startref D0: ??}%
5574 ??%
5575 }%

```

If `LWR@lateximagedepth` is known. Use a `lateximage` if the depth is greater than zero, or a regular link otherwise:

```

5576 {%
5577   \LWR@traceinfo{LWR@startref D1: \LWR@lidref}%
5578   \ifthenelse{\cnttest{\LWR@lidref}{>}{0}}%
5579   {%
5580     \LWR@traceinfo{LWR@startref D2: \LWR@lidref}%
5581     lateximage\LWR@lateximagenumberref{#1}%
5582   }%
5583   {%
5584     \LWR@traceinfo{LWR@startref D3}%
5585     #1%
5586   }%
5587 }%
5588 \LWR@traceinfo{LWR@startref E}%

```

Closing quote:

```

5589 "{}}%
5590 \LWR@traceinfo{LWR@startref F}%
5591 }

```

`\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)”.

```

5592 \NewDocumentCommand{\LWR@subnewref}{m m}{%
5593 \LWR@traceinfo{LWR@subnewref #1 #2}%
5594 \LWR@startref{#1}%
5595 \LWR@origref{#2}%
5596 \LWR@htmltag{/a}%
5597 }

```

`\ref` * `{\label}` `\ref` is `\let` to `\LWR@newref`

`\LWR@newref` * `{\label}` Create an internal document reference link, or without a link if starred per hyperref.

```

5598 \NewDocumentCommand{\LWR@newref}{s m}{%
5599 \LWR@traceinfo{LWR@newref #2}%
5600 \IfBooleanTF{#1}%
5601 {\LWR@origref{#2}}%
5602 {\LWR@subnewref{#2}{#2}}%
5603 }

```

`\pagerefPageFor` Text for page references.

```

5604 \newcommand*{\pagerefPageFor}{see }

```

`\pageref` * $\langle label \rangle$ Create an internal document reference, or just the unlinked number if starred, per `hyperref`.

```
5605 \NewDocumentCommand{\LWR@newpageref}{s m}{%
5606 \IfBooleanTF{#1}%
5607 {(\pagerefPageFor\LWR@origref{#2})}%
5608 {(\cpageref{#2})}%
5609 }
```

`\nameref` $\langle label \rangle$

```
5610 \newcommand*{\nameref}[1]{%
5611 \LWR@traceinfo{nameref A}%
5612 \LWR@startref{#1}%
5613 \LWR@traceinfo{nameref B}%
5614 \LWR@nameref{#1}%
5615 \LWR@traceinfo{nameref C}%
5616 \LWR@htmltag{/a}%
5617 \LWR@traceinfo{nameref D}%
5618 }
```

`\Nameref` $\langle label \rangle$ In print, adds the page number. In HTML, does not.

```
5619 \let\Nameref\nameref
```

59.5 Hyper-references

 Note that the code currently only sanitizes the underscore character. Additional characters should be rendered inert as well. See the `hyperref.sty` definition of `\gdef\hyper@normalise` for an example.

Pkg `hyperref`

 Do not tell other packages that `hyperref` is emulated. Some packages patch various commands if `hyperref` is present, which will probably break something, and the emulation already handles whatever may be emulated anyhow.

 Any reference to `\usepackage{hyperref}` must be placed inside a `warpprint` environment.

```
5620 % DO NOT TELL OTHER PACKAGES TO ASSUME HYPERREF, lest they attempt to patch it:
5621 % \EmulatesPackage{hyperref}[2015/08/01]% Disabled. Do not do this.
```

Create a link with a text name:

`\LWR@subhyperref` $\{\langle URL \rangle\} \{\langle text \rangle\}$

```
5622 \NewDocumentCommand{\LWR@subhyperref}{m +m}{%
5623 \LWR@htmltag{a href="#1" target="_{blank"}\LWR@orignewline}#2\LWR@htmltag{/a}%
5624 \LWR@ensuredoingapar%
5625 }
```

`\LWR@subhyperrefclass` $\{\langle URL \rangle\} \{\langle text \rangle\} \{\langle htmlclass \rangle\}$

```
5626 \NewDocumentCommand{\LWR@subhyperrefclass}{m +m m}{%
5627 \LWR@htmltag{a href="#1"
5628 class="#3"\LWR@orignewline}#2\LWR@htmltag{/a}%
5629 \LWR@ensuredoingapar%
5630 }
```

`\href` $[\langle options \rangle] \{\langle URL \rangle\} \{\langle text \rangle\}$

Create a link with accompanying text:

```
5631 \NewDocumentCommand{\LWR@hrefb}{0{} m +m}{%
5632 \LWR@subhyperref{#2}{#3}%
5633 \endgroup%
5634 \LWR@ensuredoingapar%
5635 }
5636
5637 \newcommand{\href}{%
5638 \LWR@ensuredoingapar%
5639 \begingroup%
5640 \catcode'\_ =12
5641 \LWR@hrefb%
5642 }
```

`\nolinkurl` $\{\langle URL \rangle\}$

Print the name of the link without creating the link:

```
5643 \newcommand*\LWR@nolinkurlb}[1]{#1\endgroup\LWR@ensuredoingapar}
5644
5645 \newcommand{\nolinkurl}{%
5646 \LWR@ensuredoingapar%
5647 \begingroup\catcode'\_ =12
5648 \LWR@nolinkurlb%
5649 }
```

`\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`.

```

5650 \newcommand*{\LWR@urlb}[1]{%
5651 \href{#1}{#1}%
5652 \endgroup%
5653 \LWR@ensuredoingapar%
5654 }
5655
5656 \newcommand{\LWR@urla}{%
5657 \LWR@ensuredoingapar%
5658 \begingroup\catcode'\_ =12
5659 \LWR@urlb%
5660 }
5661
5662 \let\url\LWR@urla

```

`\LWR@subinlineimage` [*alttag*] [*class*] [*filename*] [*extension*] [*style*]

```

5663 \newcommand*{\LWR@subinlineimage}[5] [] {%
5664 \ifthenelse{\equal{#1}{}}%
5665 {\LWR@htmltag{img src="#3.#4" alt="#3" style="#5" class="#2"{}}}%
5666 {\LWR@htmltag{img src="#3.#4" alt="#1" style="#5" class="#2"{}}}%
5667 }

5668 \end{warpHTML}

```

Table 9: Float data structures

For each `<type>` of float (figure, table, etc.) there exists the following:

- counter <type>:** A counter called `<type>`, such as `figure`, `table`.
 - `\<type>name`: Name. `\figurename` prints “Figure”, etc.
 - `\ext@<type>`: File extension. `\ext@figure` prints “lof”, etc.
 - `\fps@<type>`: Placement.
 - `\the<type>`: Number. `\thetable` prints the number of the table, etc.
 - `\p@<type>`: Parent’s number. Prints the number of the [within] figure, etc.
 - `\fnum@<type>`: Prints the figure number for the caption.
`\<type>name \the<type>`, “Figure 123”.
 - `\<type>`: Starts the float environment. `\figure` or `\begin{figure}`
 - `\end<type>`: Ends the float environment. `\endfigure` or `\end{figure}`
 - `\tf@<ext>`: The \TeX file identifier for the output file.
 - `LWR@have<type>`: A boolean remembering whether a `\listof` was requested for a float of this type.
 - File with extension `lo-<f,t,a-z>`:** An output file containing the commands to build the `\listof<type><name>` “table-of-contents” structure.
 - Cross-referencing names:** For `cleveref`’s `\cref` and related, `\crefname` and `\Crefname` assign human-readable names for references to this float type.
-

60 Floats

Floats are supported, although partially through emulation.

Table 9 shows the data structure associated with each `<type>` of float.

`\@makecaption` is redefined to print the float number and caption text, separated by `\CaptionSeparator`, which works with the `babel` package to adjust the caption separator according to the language. French, for example, uses an en-dash instead of a colon: “Figure 123 – Caption text”.

60.1 Float captions

for HTML output: 5669 `\begin{warpHTML}`

`\LWR@floatbegin` $\{ \langle type \rangle \} [\langle placement \rangle]$

Begins a `\newfloat` environment.

```
5670 \NewDocumentCommand{\LWR@floatbegin}{m o}{%
5671 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5672
5673 === #1 begin
5674
5675 }{}}%
5676 \LWR@stoppars
```

There is a new float, so increment the unique float counter:

```
5677 \addtocounter{\LWR@thisfloat}{1}%
5678 \booltrue{\LWR@freezethisfloat}%
```

```
5679 \begingroup
```

Settings while inside the environment:

```
5680 \LWR@origraggedright
```

Open an HTML figure tag:

```
5681 \LWR@htmltag{figure id="autofloat-\arabic{\LWR@thisfloat}" class="#1"}
```

```
5682 \renewcommand*{\@capttype}{#1}
5683 \caption@settype{#1}
5684 \LWR@startpars
5685 }
```

`\@float` Support packages which create floats directly.
`\@dblfloat`

```
5686 \let\@float\LWR@floatbegin
5687 \let\@dblfloat\LWR@floatbegin
```

`\LWR@floatend` Ends a `\newfloat` environment.

```
5688 \newcommand*{\LWR@floatend}{%
5689 \LWR@stoppars%
5690 \LWR@htmlend{figure}%
```

```

5691 \endgroup%
5692 \boolfalse{LWR@freezethisfloat}%
5693 \LWR@startpars%
5694 \ifthenelse{\boolean{FormatWordProcessor}\AND\boolean{HTMLMarkFloats}}{%
5695
5696 === end
5697
5698 }{}%
5699 }

```

`\end@float` Support packages which create floats directly.
`\end@dblfloat`

```

5700 \let\end@float\LWR@floatend
5701 \let\end@dblfloat\LWR@floatend

```

Ctr `LWR@thisfloat` A sequential counter for all floats and theorems. This is used to identify the float or theorem then reference it from the List of Figures and List of Tables.

```

5702 \newcounter{LWR@thisfloat}

```

Bool `LWR@freezethisfloat` Prevents multiple increments of `\LWR@thisfloat` inside a float.

```

5703 \newbool{LWR@freezethisfloat}
5704 \boolfalse{LWR@freezethisfloat}

```

`\LWR@maybeinthisfloat`

```

5705 \newcommand*{\LWR@maybeinthisfloat}{%
5706 \ifbool{LWR@freezethisfloat}{\addtocounter{LWR@thisfloat}{1}}%
5707 }

```

`\@capttype` Remembers which float type is in use.

```

5708 \newcommand*{\@capttype}{}

```

60.1.1 Caption inside a float environment

`\CaptionSeparator` How to separate the float number and the caption text.

```

5709 \AtBeginDocument{\providecommand*\CaptionSeparator}{:-}}

```

`\@makecaption` `{<name and num>} {<text>}`

Prints the float type and number, the caption separator, and the caption text.

```

5710 \AtBeginDocument{\renewcommand{\@makecaption}[2]{#1\CaptionSeparator#2}}

```

60.1.2 Caption and LOF linking and tracking

When a new HTML file is marked in the \LaTeX PDF file, the \LaTeX page number at that point is stored in `LWR@latestautopage`, (and the associated filename is remembered by the special \LaTeX labels). This page number is used to generate an `autofloat` HTML `<id>` in the HTML output at the start of the new HTML file. Meanwhile, there is a float counter used to generate an HTML `autofloat <id>` at the start of the float itself in the HTML file. The `autopage` and `autofloat` values to use for each float are written to the `.lof`, etc. files just before each float's entry. These values are used by `\l@figure`, etc. to create the HTML links in the List of Figures, etc.

Ctrl `LWR@nextautofloat` Tracks autofloat for floats. Tracks autopage for floats.

Ctrl `LWR@nextautopage` These are updated per float as the `.lof` file is read.

```
5711 \newcounter{LWR@nextautofloat}
5712 \newcounter{LWR@nextautopage}
```

```
\LWRsetnextfloat {<autopage>} {<autofloat>}
```

This is written to the `.lof` file just before each float's usual entry. The `autopage` and `autofloat` are remembered for `\l@figure` to use when creating the HTML links.

```
5713 \newcommand*\LWRsetnextfloat}[2]{%
5714 \setcounter{LWR@nextautopage}{#1}%
5715 \setcounter{LWR@nextautofloat}{#2}%
5716 }
```

Ctrl `LWR@latestautopage` Updated each time a new HTML file is begun. `\LWRsetnextfloat` is written with this and the `autofloat` by the modified `\addcontentsline` just before each float's entry.

```
5717 \newcounter{LWR@latestautopage}
5718 \setcounter{LWR@latestautopage}{1}

5719 \let\LWR@origcaption@begin\caption@begin
5720 \let\LWR@origcaption@end\caption@end
5721 \let\LWR@orig@@par\@par
```

`\LWR@caption@begin` Low-level patches to create HTML tags for captions.

```
5722 \newcommand{\LWR@caption@begin}
5723 {
5724 \LWR@traceinfo{LWR@caption@begin}%
```

Keep `par` and `minipage` changes local:

```
5725 \begingroup%
```

The caption code was not allowing the closing par tag:

```
5726 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@par}%
```

No need for a minipage or \parbox inside the caption:

```
5727 \RenewDocumentEnvironment{minipage}{O{t} o O{t} m}{-}{-}%
5728 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}{##5}%
```

Enclose the original caption code inside an HTML tag:

```
5729 \LWR@htmlblocktag{figcaption}%
5730 \LWR@origcaption@begin%
5731 }
```

`\LWR@caption@end` Low-level patches to create HTML tags for captions.

```
5732 \newcommand{\LWR@caption@end}
5733 {%
5734 \LWR@origcaption@end%
```

Subcaptions were being over-written by the closing HTML tag:

```
5735 \LWR@origvspace*{\baselineskip}%
```

Closing tag:

```
5736 \LWR@htmlblocktag{/figcaption}%
5737 \endgroup%
5738 % \leavevmode% avoid bad space factor (0) error
5739 \LWR@traceinfo{\LWR@caption@end: done}%
5740 }
```

`\caption@begin` Low-level patches to create HTML tags for captions.
`\caption@end`

```
5741 \AtBeginDocument{
5742 \let\caption@begin\LWR@caption@begin
5743 \let\caption@end\LWR@caption@end
5744 }
```

`\captionlistentry` Tracks the float number for this caption used outside a float. Patched to create an HTML anchor.

```
5745 \let\LWR@origcaptionlistentry\captionlistentry
5746
5747 \renewcommand*{\captionlistentry}{%
```

```

5748 \LWR@maybeinthisfloat%
5749 \LWR@ensuredoingapar%
5750 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5751 \LWR@origcaptionlistentry%
5752 }
5753
5754 \def\LWR@LTcaptionlistentry{%
5755 \LWR@ensuredoingapar%
5756 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5757 \bgroup
5758 \@ifstar{\egroup\LWR@LTcaptionlistentry}% gobble *
5759 {\egroup\LWR@LTcaptionlistentry}}%
5760 \def\LWR@LTcaptionlistentry#1{%
5761 \caption@listentry\@firstoftwo[LTcaptype]{#1}}%

```

`\addcontentsline` Patched to write the autopage and autofloat before each float's entry. No changes if writing `.toc` For a theorem, automatically defines `\ext@<type>` as needed, to mimic and reuse the float mechanism.

```

5762 \let\LWR@origaddcontentsline\addcontentsline
5763
5764 \renewcommand*{\addcontentsline}[3]{%
5765 \ifthenelse{\equal{#1}{toc}}{ }{%
5766 \ifthenelse{\equal{#1}{thm}}{\csdef{ext@#2}{thm}}{ }
5767 \addtocontents{\@nameuse{ext@#2}}{%
5768 \protect\LWRsetnextfloat%
5769 {\arabic{LWR@latestautopage}}%
5770 {\arabic{LWR@thisfloat}}%
5771 }% addtocontents
5772 }% not toc
5773 \LWR@origaddcontentsline{#1}{#2}{#3}%
5774 }

```

`\captionof` Patched to track the float number since this is used outside a float, and also create an HTML anchor for the virtual float.

```

5775 \AtBeginDocument{
5776 \let\LWR@origcaptionof\captionof
5777
5778 \renewcommand*{\captionof}{%
5779 \LWR@maybeinthisfloat%
5780 \LWR@stoppars
5781 \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
5782 \LWR@origcaptionof%
5783 }
5784 }

5785 \end{warpHTML}

```

61 Table of Contents, LOF, LOT

This section controls the generation of the TOC, LOF, and LOT.

The `.toc`, `.lof`, and `.lot` files are named by the source code `\jobname`.

In HTML, the printed tables are placed inside a `<div>` of class `toc`, `lof`, or `lot`.

A “sideroc” 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: 5786 `\begin{warpHTML}`

61.1 Reading and printing the TOC

`\LWR@myshorttoc` `{\{toc/lof/lot\}}`

Reads in and prints the TOC/LOF/LOT at the current position. While doing so, makes the @ character into a normal letter to allow formatting commands in the section names.

Unlike in regular \TeX , the file is not reset after being read, since the TOC may be referred to again in each HTML page, and is used for the sideroc.

```
5787 \newcommand*{\LWR@myshorttoc}[1]{
5788 \LWR@ensuredoingapar
```

Only if the file exists:

```
5789 \IfFileExists{\jobname.#1}{
```



Make @ a regular letter. Many of the commands in the file will have @ characters in them, so @ must be made a regular letter.



For `pdflatex`, also change to `latin1` encoding. When reading back a file with accented characters, the encoding change seems to be required, rather than leaving it `utf8`.

```
5790 \begingroup
5791 % \ifxetexorluatex%
5792 % \else
5793 % \inputencoding{latin1}% currently disabled
5794 % \fi
```

5795 \makeatletter

Read in the TOC file:

```
5796 \@input{\jobname.#1}
5797 % \makeatother
5798 \endgroup
5799 }%
5800 {}%
5801 }
```

`\LWR@subtableofcontents` $\{ \langle toc/lof/lot \rangle \} \{ \langle sectionstarname \rangle \}$

Places a TOC/LOF/LOT at the current position.

```
5802 \NewDocumentCommand{\LWR@subtableofcontents}{m m}{%
```

Closes previous levels:

```
5803 \@ifundefined{chapter}
5804 {\LWR@closeprevious{\LWR@depthsection}}
5805 {\LWR@closeprevious{\LWR@depthchapter}}
```

Prints any pending footnotes so that they appear above the potentially large TOC:

```
5806 \LWR@printpendingfootnotes
```

Place the list into its own chapter (if defined) or section:

```
5807 \@ifundefined{chapter}{\section*{#2}}{\chapter*{#2}}
```

Create a new HTML nav containing the TOC/LOF/LOT:

```
5808 \LWR@htmlclass{nav}{#1}
```

Create the actual list:

```
5809 \LWR@myshorttoc{#1}
```

Close the nav:

```
5810 \LWR@htmlclassend{nav}{#1}
5811 }
```

Patch `\starttoc` to encapsulate the TOC inside HTML tags:

```
5812 \let\LWR@orig@starttoc\starttoc
```

```

5813
5814 \renewcommand{\@starttoc}[1]{
5815 \LWR@htmlclass{nav}{#1}
5816 \LWR@orig@starttoc{#1}
5817 \LWR@htmlclassend{nav}{#1}
5818 }

```

Patch `\tableofcontents`, etc. to print footnotes first. `newfloat` uses `\listoffigures` for all future float types.

```

5819 \let\LWR@origtableofcontents\tableofcontents
5820 \let\LWR@origlistoffigures\listoffigures
5821 \let\LWR@origlistoftables\listoftables
5822
5823 \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:

```

5824 \ifbool{FormatWordProcessor}{}{

```

Copy the `.toc` file to `.sideroc` for printing the sideroc. The original `.toc` file is renewed when `\tableofcontents` is finished.

```

5825   \LWR@copyfile{\jobname.toc}{\jobname.sideroc}%
5826   \LWR@printpendingfootnotes
5827   \LWR@origtableofcontents
5828 }
5829 }
5830 \renewcommand*{\listoffigures}{
5831 \ifbool{FormatWordProcessor}{}{
5832   \LWR@printpendingfootnotes
5833   \LWR@origlistoffigures
5834 }
5835 }
5836
5837 \renewcommand*{\listoftables}{
5838 \ifbool{FormatWordProcessor}{}{
5839   \LWR@printpendingfootnotes
5840   \LWR@origlistoftables
5841 }
5842 }

```

61.2 High-level TOC commands

```
\listof {<type>} {<title>}
```

Emulate the `\listof` command from the float package (section 128). Used to create lists of custom float types. Also used to redefine the standard \TeX `\listoffigures` and `\listoftables` commands.

```
5843 \NewDocumentCommand{\listof}{m +m}{%
5844 \LWR@subtableofcontents{\@nameuse{ext@#1}}{#2}
5845 \expandafter\newwrite\csname tf@\csname ext@#1\endcsname\endcsname
5846 \immediate\openout \csname tf@\csname ext@#1\endcsname\endcsname
5847   \jobname.\csuse{ext@#1}\relax
5848 }
```

61.3 Side TOC

The “side TOC” is a table-of-contents positioned to the side.

It may be renamed by redefining `\sitetocname`, 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.

```
5849 \end{warpHTML}
```

for HTML & PRINT: 5850 `\begin{warpall}`

Ctrl `SideTOCDepth` Controls how deep the side-TOC gets. Use a standard \TeX section level similar to `tocdepth`.

```
5851 \newcounter{SideTOCDepth}
5852 \setcounter{SideTOCDepth}{1}
```

`\sitetocname` Holds the default name for the sideroc.

```
5853 \newcommand{\sitetocname}{Contents}
```

```
5854 \end{warpall}
```

for HTML output: 5855 `\begin{warpHTML}`

`\LWR@sidetoc` Creates the actual side-TOC.

```
5856 \newcommand*{\LWR@sidetoc}{
5857 \LWR@forcenewpage
5858 \LWR@stoppars
5859 }
```

The entire sidetoc is placed into a `nav` of class `sidetoc`.

```
5860 \LWR@htmlclass{nav}{sidetoc}
5861
5862 \setcounter{tocdepth}{\value{SideTOCDepth}}
5863 }
```

The title is placed into a `<div>` of class `sidetoctitle`, and may contain paragraphs.

```
5864 \begin{BlockClass}{sidetoctitle}
5865 \sidetocname
5866 \end{BlockClass}
```

The table of contents is placed into a `<div>` of class `sidetoccontents`.

```
5867 \begin{BlockClass}{sidetoccontents}
5868 \LinkHome
5869
5870 \LWR@myshorttoc{sidetoc}
5871 \end{BlockClass}
5872 \LWR@htmlclassend{nav}{sidetoc}
5873 }
```

61.4 Low-level TOC line formatting

`\numberline` `{\langle number \rangle}`

(Called from each line in the `.aux`, `.lof` files.)

Record this section number for further use:

```
5874 \renewcommand*{\numberline}[1]{%
5875 \LWR@sectionnumber{#1}\quad%
5876 }
```

`\hypertoc` `{\langle 1: depth \rangle} {\langle 2: type \rangle} {\langle 3: name \rangle} {\langle 4: page \rangle}`

Called by `\l@section`, etc. to create a hyperlink to a section.

The autopage label is always created just after the section.

#1 is depth

#2 is section, subsection, etc.

#3 the text of the caption

#4 page number

```
5877 \NewDocumentCommand{\hypertoc}{m m +m m}{%
```

Respond to tocdepth:

```
5878 \ifthenelse{\cinttest{#1}{<=} {\value{tocdepth}}}{%
5879   \LWR@startpars%
```

Create an HTML link to filename#autosec-(page), with text of the caption, of the given HTML class.

```
5880   \LWR@subhyperrefclass{%
5881   \LWR@htmlrefsectionfilename{autopage-#4}\#autosec-#4}{#3}{toc#2}%
5882   \LWR@stoppars%
5883 }
5884 {}
5885 }
```

Ctrl lofdepth TOC depth for figures.

```
5886 \newcounter{lofdepth}
5887 \setcounter{lofdepth}{1}
```

Ctrl lotdepth TOC depth for tables.

```
5888 \newcounter{lotdepth}
5889 \setcounter{lotdepth}{1}
```

```
\hypertocfloat {<1: depth>} {<2: type>} {<3: ext of parent>} {<4: caption>} {<5: page>}
```

#1 is depth

#2 is figure, table, etc.

#3 is lof, lot, of the parent.

#4 the text of the caption

#5 page number

```
5890 \newcommand{\hypertocfloat}[5]{%
5891 \LWR@startpars
```

If some float-creation package has not yet defined the float type's `lofdepth` counter, etc, define it here:

```
5892 \@ifundefined{c@#3depth}{%
5893 \newcounter{#3depth}%
5894 \setcounter{#3depth}{1}%
5895 }{}%
```

Respond to `lofdepth`, etc.:

```
5896 \LWR@traceinfo{hypertocfloat depth is #1 #3depth is \arabic{#3depth}}%
5897 \ifthenelse{\cnttest{#1}<=}{\arabic{#3depth}}{}%
5898 \LWR@startpars%
```

Create an HTML link to `filename#autofloat-`(float number), with text of the caption, of the given HTML class.

```
5899 \LWR@subhyperrefclass{%
5900 \LWR@htmlrefsectionfilename{autopage-\arabic{LWR@nextautopage}}%
5901 \#autofloat-\arabic{LWR@nextautofloat}}%
5902 {#4}{toc#2}%
5903 \LWR@stoppars%
5904 }{}%
5905 }
```

Automatically called by `\contentsline`:

```
5906 \renewcommand{\l@part}[2]{\hypertoc{-1}{part}{#1}{#2}}
5907 \DeclareDocumentCommand{\l@chapter}{m m}
5908   {\hypertoc{0}{chapter}{#1}{#2}}
5909 \renewcommand{\l@section}[2]{\hypertoc{1}{section}{#1}{#2}}
5910 \renewcommand{\l@subsection}[2]{\hypertoc{2}{subsection}{#1}{#2}}
5911 \renewcommand{\l@subsubsection}[2]
5912   {\hypertoc{3}{subsubsection}{#1}{#2}}
5913 \renewcommand{\l@paragraph}[2]{\hypertoc{4}{paragraph}{#1}{#2}}
5914 \renewcommand{\l@subparagraph}[2]{\hypertoc{5}{subparagraph}{#1}{#2}}
5915 \renewcommand{\l@figure}[2]{\hypertocfloat{1}{figure}{lof}{#1}{#2}}
5916 \renewcommand{\l@table}[2]{\hypertocfloat{1}{table}{lot}{#1}{#2}}

5917 \end{warpHTML}
```

62 Index and glossary

See:

<http://tex.stackexchange.com/questions/187038/>

[how-to-mention-section-number-in-index-created-by-imakeidx](#)

Index links are tracked by the counter `LWR@autoindex`. This counter is used to create a label for each index entry, and a reference to this label for each entry in the index listing. This method allows each index entry to link directly to its exact position in the document.

for HTML output: 5918 `\begin{warpHTML}`

```
5919 \newcounter{LWR@autoindex}
5920 \setcounter{LWR@autoindex}{0}
5921
5922 \newcounter{LWR@autoglossary}
5923 \setcounter{LWR@autoglossary}{0}
```

`\printindex`

```
5924 \let\LWR@origprintindex\printindex
5925
5926 \renewcommand*\printindex
5927 {
5928 \LWR@startpars
5929 \LWR@origprintindex
5930 }
```

Env `theindex`

```
5931 \@ifundefined{chapter}
5932 {\newcommand*\LWR@indexsection}[1]{\section*{#1}}
5933 {\newcommand*\LWR@indexsection}[1]{\chapter*{#1}}
5934
5935 \renewenvironment*{theindex}{%
5936 \LWR@indexsection{\indexname}%
5937 \let\item\LWR@indexitem%
5938 \let\subitem\LWR@indexsubitem%
5939 \let\subsubitem\LWR@indexsubsubitem%
5940 }{}
```

`\LWR@indexitem`

```
5941 \newcommand*\LWR@indexitem){
5942
```

```
5943 \InlineClass{indexitem}{  
5944 }
```

\LWR@indexitem

```
5945 \newcommand{\LWR@indexsubitem}{  
5946  
5947 \InlineClass{indexsubitem}{  
5948 }
```

\LWR@indexitem

```
5949 \newcommand{\LWR@indexsubsubitem}{  
5950  
5951 \InlineClass{indexsubsubitem}{  
5952 }
```

\@wrindex {*{term}*} Redefined to write the LWR@latestautopage counter instead of page

```
5953 \def \@wrindex#1{%  
5954 \addtocounter{LWR@autoindex}{1}%  
5955 \LWR@newlabel{LWRindex-\theLWR@autoindex}%  
5956 \protected@write\@indexfile{%  
5957 {\string\indexentry{#1}{\theLWR@autoindex}}%  
5958 \endgroup  
5959 \@esphack}
```

\@wrglossary {*{term}*} Redefined to write the LWR@latestautopage counter instead of page

```
5960 \def \@wrglossary#1{%  
5961 \addtocounter{LWR@autoglossary}{1}%  
5962 \LWR@newlabel{LWRglossary-\theLWR@autoglossary}%  
5963 \protected@write\@glossaryfile{%  
5964 {\string\glossaryentry{#1}{\theLWR@autoglossary}}%  
5965 \endgroup  
5966 \@esphack}
```

\hyperindexref {*{autosecnumber}*}

\hyperindexref{web address} is inserted into *.ind by the xindy style file lwarp.xdy

```
5967 \newcommand*{\hyperindexref}[1]{\nameref{LWRindex-#1}}  
  
5968 \end{warpHTML}
```

for PRINT output: A null command for print mode, in case hyperref was not used:

```
5969 \begin{warpprint}
5970 \newcommand{\hyperindexref}[1]{#1}
5971 \end{warpprint}
```

for HTML & PRINT: For the glossaries package, try to prevent an error where `\glo@name` was not found:

```
5972 \begin{warppall}
5973 \providecommand{\glo@name}{}
5974 \end{warppall}
```

63 Math

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

Math rendering Math may be rendered as SVG graphics or using the MATHJAX JavaScript display engine.

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

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

SVG files As currently implemented, each instance of math creates a new SVG file. In text with many references to math variables, this can result in a large number of files with duplicate content. In the future, some method of content-based naming and checksumming may be used to remove the need for duplicate files.

SVG inline Another approach would be to in-line the SVG files directly into the HTML. This avoids having a large number of files and potentially speeds loading the images, but dis-allows the possibility of sharing one file among many instances without user intervention.

¹³See section 240 regarding fonts and fractions.

PNG files Others have used PNG files, sometimes pre-scaled for print resolution but displayed on-screen at a scaled down size. This allows high-quality print output at the expense of larger files, but SVG files are also larger as well.

MathML Conversion to MathML might be a better approach, among other things allowing a more compact representation of math than SVG drawings. Problems with MathML include limited browser support and some issues with the fine control of the appearance of the result. Also see section 8 regarding EPUB output with MathJax.

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

Prog MathJax

When MathJax is enabled, math is rendered twice:

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

MathJax limitations Limitations when using MathJax include:

Prog MathJax

chapter numbers

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

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

subequations

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

footnotes in math

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

lateximage

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

siunitx

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

<https://github.com/mathjax/MathJax-third-party-extensions/tree/master/siunitx>

 **siunitx inside an equation**

LaTeX macros

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

custom MathJax macros

As an example of using custom LaTeX macros with MathJax, place the following at the start of the document, after `\begin{document}`:

```
\begin{warpHTML} % Only for HTML output,
\ifbool{mathjax} % and only for MathJax output:
{
\(\ % New macros for MathJax are placed inside a math expression:
\newcommand{\expval}[1]{\langle#1\rangle}
\newcommand{\abs}[1]{\lvert#1\rvert}
\)
}{
}
\end{warpHTML}
```

for HTML output: 5975 `\begin{warpHTML}`

- \\$** Plain dollar signs appearing in the HTML output may be interpreted by MathJax to be math shifts. For a plain text dollar `\$`, print it inside a span to avoid it being interpreted by MathJax, unless are inside a `lateximage`, in which case it will not be seen by MathJax.

```
5976 \let\LWR@origtextdollar\$
5977
5978 \renewcommand*{\$}{%
5979 \ifthenelse{\cinttest{\value{LWR@lateximagedepth}}{>}{0}}{
5980 {\LWR@origtextdollar}%
5981 {\LWR@htmltagc{span}\LWR@origtextdollar\LWR@htmltagc{/span}}%
5982 }
```

ctr `LWR@externalfilecnt` Counter for the external files which are generated and then referenced from the HTML:

```
5983 \newcounter{LWR@externalfilecnt}
```

63.1 Inline and display math

```
5984 \LetLtxMacro\LWR@origdollar$
5985 \LetLtxMacro\LWR@secondorigdollar$% balance for editor syntax highlighting

5986 \LetLtxMacro\LWR@origopenparen\(  
5987 \LetLtxMacro\LWR@origcloseparen\)
```

```
5988 \LetLtxMacro\LWR@origopenbracket\[
5989 \LetLtxMacro\LWR@origclosebracket\]
```

\$ Redefine the dollar sign to place math inside a lateximage, or use MathJax:
\$\$

```
5990 \begingroup
5991 \catcode'\$=\active%
5992 \protected\gdef$\@ifnextchar$\LWR@doubledollar\LWR@singledollar}%
```

\LWR@doubledollar Redefine the double dollar sign to place math inside a lateximage, or use MathJax:

```
5993 \gdef\LWR@doubledollar$#1$${
5994 \ifbool{mathjax}%
```

For MathJax, print the math between $\[$ and $\]$:

```
5995 {\textbackslash[\LWR@HTMLsanitize{#1}\textbackslash]}
```

For SVG, print the math inside a lateximage, with an ALT tag of the \LaTeX code:

```
5996 {% not mathjax
5997
5998 \begin{lateximage}%
5999 [\textbackslash[[] \LWR@HTMLsanitize{#1} \textbackslash[]]}%
6000 \LWR@origdollar\LWR@origdollar#1\LWR@origdollar\LWR@origdollar%
6001 \end{lateximage}
6002
6003 }
6004 }%
```

\LWR@singledollar Redefine the single dollar sign to place math inside a lateximage, or use MathJax:

```
6005 \gdef\LWR@subsingledollar#1{%
6006 \ifbool{mathjax}%
```

For MathJax, print the math between $\($ and $\)$:

```
6007 {\textbackslash(\LWR@HTMLsanitize{#1}\textbackslash)}
```

For SVG, print the math inside a lateximage, with an ALT tag of the \LaTeX code:

```
6008 {% not mathjax
6009 \begin{lateximage}%
6010 [\textbackslash(\LWR@HTMLsanitize{#1} \textbackslash)]%
6011 \LWR@origdollar#1\LWR@origdollar%
6012 \end{lateximage}%
6013 }%
6014 }
```

```

6015
6016 \gdef\LWR@singledollar#1${%
6017 \LWR@subsingledollar{#1}%
6018 }

```

\(Redefine to the above dollar macros.
 \)

```

6019 \gdef\(#1\){$#1$}
6020 \gdef\[#1\]{$$#1$$}
6021
6022 \endgroup
6023
6024
6025

```

\@ensuredmath $\{expression\}$ Not yet adapted to lwarp.

```

6026 \let\LWR@orig@ensuredmath\@ensuredmath
6027
6028 \newcommand*\LWR@ensuredmath[1]{%
6029 \LWR@traceinfo{\@ensuredmath: page \thepage: \detokenize{#1}}%
6030 \LWR@subsingledollar{#1}%
6031 }

```

Remove the old math and displaymath environments:

```

6032 \let\math\relax
6033 \let\endmath\relax
6034 \let\displaymath\relax
6035 \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`.

```

6036 \NewEnviron{math}{\expandafter\(\BODY\)}

```

Env `displaymath` Set math mode then typeset the body of what was between the begin/end. See the `environ` package for `\BODY`.

```

6037 \NewEnviron{displaymath}{\expandafter\[\BODY\]\@ignoretrue}

```

63.2 MathJax support

Ctrl `LWR@nextequation` Used to add one to compute the next equation number.

```
6038 \newcounter{LWR@nextequation}
```

`\LWR@syncmathjax` Sets the MathJax equation format and number for the following equations.

These MathJax commands are printed inside “\(" and “\)” characters. They are printed to HTML output, not interpreted by \LaTeX .

```
6039 \newcommand*{\LWR@syncmathjax}{%
```

If using chapters, place the chapter number in front of the equation. Otherwise, use the simple equation number.

```
6040 \ifcsdef{thechapter}{
6041 \InlineClass{hidden}{
6042 \textbackslash(
6043 \textbackslash{}seteqsection \{\thechapter\}
6044 \textbackslash)
6045 }
6046 }
6047 {}}% not using chapters
```

MathJax doesn't allow setting the equation number to 1:

```
6048 \ifthenelse{\cnttest{\value{equation}}>0}
6049 {
```

Tell MathJax that the next set of equations begins with the current \LaTeX equation number, plus one.

```
6050 \setcounter{LWR@nextequation}{\value{equation}}
6051 \addtocounter{LWR@nextequation}{1}
```

Place the MathJax command inside “\(" and “\)” characters, to be printed to HTML, not interpreted by \LaTeX .

```
6052 \InlineClass{hidden}{
6053 \textbackslash(
6054 \textbackslash{}seteqnumber \{\arabic{LWR@nextequation}\}
6055 \textbackslash)
6056 }
6057 {}}% not eq > 0
6058 }
```

`\LWR@restoreorigformatting` While producing SVG math or a `lateximage`, use regular \LaTeX formatting instead of HTML tags.

Various packages add to this macro using `\appto`.

```
6059 \newcommand*{\LWR@restoreorigformatting}{%
6060 \LWR@traceinfo{\LWR@restoreorigformatting}%
6061 \RenewDocumentCommand{\InlineClass}{o m +m}{##3}%
6062 \RenewDocumentEnvironment{BlockClass}{o m}{}{}%
6063 \renewcommand{\BlockClassSingle}[2]{##2}%
6064 \LetLtxMacro{\hspace}{\LWR@orighspace}%
6065 \LetLtxMacro{\rule}{\LWR@origrule}%
6066 \let\,\LWR@origcomma% disable HTML short unbreakable space
6067 \let\textellipsis\LWR@origtextellipsis%
6068 \let\textless\LWR@origtextless%
6069 \let\textgreater\LWR@origtextgreater%
6070 \LetLtxMacro{\textrm}{\LWR@origtextrm}%
6071 \LetLtxMacro{\textsf}{\LWR@origtextsf}%
6072 \LetLtxMacro{\texttt}{\LWR@origtexttt}%
6073 \LetLtxMacro{\textbf}{\LWR@origtextbf}%
6074 \LetLtxMacro{\textmd}{\LWR@origtextmd}%
6075 \LetLtxMacro{\textit}{\LWR@origtextit}%
6076 \LetLtxMacro{\textsl}{\LWR@origtextsl}%
6077 \LetLtxMacro{\textsc}{\LWR@origtextsc}%
6078 \LetLtxMacro{\textup}{\LWR@origtextup}%
6079 \LetLtxMacro{\textnormal}{\LWR@origtextnormal}%
6080 \LetLtxMacro{\emph}{\LWR@origemph}%
6081 \LetLtxMacro{\rmfamily}{\LWR@origrmfamily}%
6082 \LetLtxMacro{\sffamily}{\LWR@origsffamily}%
6083 \LetLtxMacro{\ttfamily}{\LWR@origttfamily}%
6084 \LetLtxMacro{\bfseries}{\LWR@origbfseries}%
6085 \LetLtxMacro{\mdseries}{\LWR@origmdseries}%
6086 \LetLtxMacro{\upshape}{\LWR@origupshape}%
6087 \LetLtxMacro{\slshape}{\LWR@origslshape}%
6088 \LetLtxMacro{\scshape}{\LWR@origscshape}%
6089 \LetLtxMacro{\itshape}{\LWR@origitshape}%
6090 \LetLtxMacro{\em}{\LWR@origem}%
6091 \LetLtxMacro{\normalfont}{\LWR@orignormalfont}%
6092 \let\sp\LWR@origsp%
6093 \let\sb\LWR@origsb%
6094 \let\textsuperscript\LWR@origtextsuperscript%
6095 \let\textsubscript\LWR@origtextsubscript%
6096 \LetLtxMacro{\underline}{\LWR@origunderline}%
6097 \let~\LWR@origtilde%
6098 \let\enskip\LWR@origenskip%
6099 \let\quad\LWR@origquad%
6100 \let\qqquad\LWR@origqqquad%
6101 \LetLtxMacro{\tabular}{\LWR@origtabular}%
6102 \LetLtxMacro{\endtabular}{\LWR@origendtabular}%
6103 \LetLtxMacro{\toprule}{\LWR@origtoprule}%
6104 \LetLtxMacro{\midrule}{\LWR@origmidrule}%
6105 \LetLtxMacro{\cmidrule}{\LWR@origcmidrule}%
6106 \LetLtxMacro{\bottomrule}{\LWR@origbottomrule}%
6107 \LetLtxMacro{\addlinespace}{\LWR@origaddlinespace}%
6108 \LetLtxMacro{\morecmidrules}{\LWR@origmorecmidrules}
```

```

6109 \LetLtxMacro\specialrule\LWR@origspecialrule%
6110 \let\newline\LWR@orignewline%
6111 \LetLtxMacro{\raisebox}{\LWR@origraisebox}%
6112 \LetLtxMacro\includegraphics\LWR@origincludegraphics%
6113 \LetLtxMacro{\scalebox}{\LWR@origscalebox}%
6114 \LetLtxMacro{\rotatebox}{\LWR@origrotatebox}%
6115 \let\reflectbox\LWR@origreflectbox%
6116 \LetLtxMacro\resizebox\LWR@origresizebox%
6117 \let\framebox\LWR@origframebox%
6118 \let\makebox\LWR@origmakebox%
6119 \let\fbbox\LWRprint@fbbox%
6120 \let\fbboxBlock\LWRprint@fbbox%
6121 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}%
6122 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}%
6123 \LetLtxMacro{\minipage}{\LWR@origminipage}%
6124 \let\endminipage\LWR@endminipage%
6125 \LetLtxMacro{\parbox}{\LWR@origparbox}%
6126 \let\TeX\LWR@origTeX%
6127 \let\LaTeX\LWR@origLaTeX%
6128 \let\LaTeXe\LWR@origLaTeXe%
6129 \renewcommand*{\Xe}{X\textsubscript{E}}%
6130 %
6131 \LWR@restoreorigaccents%
6132 %
6133 \LWR@FBcancel%
6134 }

```

`\LWR@hidelatexequation` $\{ \langle environment \rangle \} \{ \langle contents \rangle \}$

Creates the \LaTeX version of the equation inside an HTML comment.

```
6135 \NewDocumentCommand{\LWR@hidelatexequation}{m +m}{%
```

Stop HTML paragraph handling and open an HTML comment:

```

6136 \LWR@stoppars
6137 \LWR@htmlopencomment
6138

```

Start the \LaTeX math environment inside the HTML comment:

```

6139 \begingroup
6140 \csuse{LWR@orig#1}

```

While in the math environment, restore various commands to their \LaTeX meanings.

```
6141 \LWR@restoreorigformatting
```

See `\LWR@htmlmathlabel` in section 63.4.1.

Print the contents of the equation:

```
6142 #2
```

End the \LaTeX math environment inside the HTML comment:

```
6143 \csuse{LWR@origend#1}
6144 \endgroup
6145
```

Close the HTML comment and resume HTML paragraph handling:

```
6146 \LWR@htmlclosecomment
6147 \LWR@startpars
6148 }
```

```
\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 .

```
6149 \NewDocumentCommand{\LWR@addmathjax}{m +m}{%
```

Enclose the MathJax environment inside printed “\ (“ and “\)” characters.

```
6150 \LWR@origtilde\LWR@orignewline
6151 \textbackslash{}begin\{#1\}
```

Print the contents, sanitizing for HTML special characters.

```
6152 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{#2}}
```

Close the MathJax environment:

```
6153 \textbackslash{}end\{#1\}
6154 \LWR@orignewline
6155 }
```

63.3 Equation environment

Remember existing equation environment:

```
6156 \let\LWR@origequation\equation
6157 \let\LWR@origendequation\endequation
```

Remove existing equation environment:

```
6158 \let\equation\relax
6159 \let\endequation\relax
```

Env `equation` The new equation environment is created with `\NewEnviron` (from the `environ` package), which stores the contents of its environment in a macro called `\BODY`.

For SVG math output, the contents are typeset using the original `equation` inside a `lateximage`, along with an ALT tag containing a detokenized copy of the \TeX source for the math.

For MathJax output, the contents are typeset in an original `equation` environment placed inside a HTML comment, with special processing for `\labels`. The contents are also printed to the HTML output for processing by the MathJax script.

```
6160 \NewEnviron{equation}{%
6161
6162 \ifbool{mathjax}
```

MathJax output:

```
6163 {
```

Print commands to synchronize MathJax's equation number and format to the current \TeX chapter/section and equation number:

```
6164 \LWR@syncmathjax
```

Print the \TeX math inside an HTML comment:

```
6165 \LWR@hidelatexequation{equation}{\BODY}
6166 }
```

SVG output: Create the `lateximage` along with an HTML ALT tag having an equation number, the \TeX equation environment commands, and the contents of the environment's `\BODY`.

```
6167 {% not mathjax
```

Begin the `lateximage` with an ALT tag containing the math source:

```
6168 \begin{lateximage}[{\theequation} \textbackslash{begin\{equation\}}]%
6169 \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\BODY}}%
6170 \textbackslash{end\{equation\}}]% alt tag
```

Create the actual \LaTeX -formatted equation inside the `lateximage` using the contents of the environment.

```
6171 \LWR@origequation
6172 \BODY% contents collected by NewEnviron
6173 \LWR@origendequation
6174 \end{lateximage}%
6175 }
6176
```

After the environment, if MathJax, print the math to the HTML output for MathJax processing:

```
6177 }[\ifbool{mathjax}{\LWR@addmathjax{equation}{\BODY}}{}]
```

63.4 AMS Math environments

63.4.1 Support macros

Bool `LWR@amsmultline` True if processing a multiline environment.

To compensate for `multline`-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.

```
6178 \newbool{LWR@amsmultline}
6179 \boolfalse{LWR@amsmultline}
```

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

`lwarp` points `\ltx@label` here. This is used by `\label` when inside a \LaTeX AMS math environment's math display environment.

`\LWR@origltx@label` points to the \LaTeX original, modified by `lwarp`, then by `amsmath`, then by `cleveref`.

```
6180 \newcommand*{\LWR@htmlmathlabel}[1]{%
6181 \LWR@traceinfo{\LWR@htmlmathlabel #1}%
6182 \ifbool{mathjax}{%
```

The combined \LaTeX & HTML label is printed in a `\text` field:

```
6183 \text{
```

Shift the label over to the right side of the environment to avoid over-printing the math:

```
6184 \ifbool{LWR@amsmultiline}{\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.)

```
6185 \LWR@htmlclosecomment%
6186 \LWR@origltx@label{#1}%
6187 \LWR@htmlopencomment%
6188 }% text
6189 }% mathjax
6190 {%
6191 \LWR@origltx@label{#1}%
6192 }%
6193 }
```

`\LWR@beginhideamsmath` Starts hiding \LaTeX math inside an HTML comment.

```
6194 \newcommand*{\LWR@beginhideamsmath}{
6195 \LWR@stoppars
6196 \LWR@origtilde\LWR@orignewline
6197 \LWR@htmlopencomment
6198
6199 \begingroup
6200 \LWR@restoreorigformatting
6201 }
```

`\LWR@endhideamsmath` Ends hiding \LaTeX math inside an HTML comment.

```
6202 \newcommand*{\LWR@endhideamsmath}{
6203 \endgroup
6204
6205 \LWR@htmlclosecomment
6206 \LWR@orignewline
6207 \LWR@startpars
6208 }
```

63.4.2 Environment patches

The following `amsmath` environments already collect their contents in `\@envbody` for further processing. `eqnarray` is not an \mathcal{AMS} package, and thus requires special handling.

For `svg math`: Each environment is encapsulated inside a `lateximage` environment, along with a special `LWRAMSMATHBODY` argument telling `lateximage` to use as the `HTML ALT` tag the environment's contents which were automatically captured by the \mathcal{AMS} environment.

For `MathJax`: Each environment is syched with \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`.

```
6209 \let\LWR@origeqnarray\eqnarray
6210 \let\LWR@origendeqnarray\endeqnarray
```

To remember whether the starred environment was used, and thus whether to number the equations:

```
6211 \newbool{LWR@numbereqnarray}
6212 \booltrue{LWR@numbereqnarray}
```

Common code used by `eqnarray` and `Beqnarray` (from `fancybox`):

```
6213 \newcommand{\LWR@eqnarrayfactor}{%
```

Different behavior depending on `MathJax` vs `svg math`:

```
6214 \ifbool{mathjax}
6215 {
```

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:

```
6216   \LWR@syncmathjax
6217   \boolfalse{LWR@amsmultiline}
6218   \ifbool{LWR@numbereqnarray}
6219   {
```

If numbering the equations, execute a copy inside an `HTML` comment block:

```
6220   \LWR@beginhideamsmath
6221   \LWR@origeqnarray
6222   \BODY
6223   \LWR@origendeqnarray
6224   \LWR@endhideamsmath
```

Then print the (sanitized) contents to the output for MathJax to interpret:

```
6225     \LWR@addmathjax{eqnarray}{\BODY}
6226     }%
6227     {% not LWR@numbreqnarray
```

If not numbering equations, just create the contents for MathJax:

```
6228     \LWR@addmathjax{eqnarray*}{\BODY}
6229     }% LWR@numbreqnarray
6230 }% mathjax
6231 {% not mathjax
6232     \ifbool{LWR@numbreqnarray}
6233     {
```

For numbered svg equations, first create a lateximage with an alt attribute containing sanitized copy of the source code:

```
6234     \begin{lateximage}[\LWR@addmathjax{eqnarray}{\BODY}]
```

Then create the image contents using an actual eqnarray:

```
6235     \LWR@origeqnarray
6236     \BODY
6237     \LWR@origendeqnarray
6238     \end{lateximage}
6239     }%
6240     {% not LWR@numbreqnarray
```

If not numbered, do the same, but an extra `\nonumber` seems to be required:

```
6241     \begin{lateximage}[\LWR@addmathjax{eqnarray*}{\BODY}]
6242     \LWR@origeqnarray
6243     \BODY
6244     \nonumber
6245     \LWR@origendeqnarray
6246     \end{lateximage}
6247     }% LWR@numbreqnarray
6248 }% not mathjax
```

Default to number equations in the future:

```
6249 \booltrue{LWR@numbreqnarray}
6250 }
```

`eqnarray` itself is made with a blank line before and after to force it to be on its own line:

```

6251 \RenewEnviron{eqnarray}
6252 {%
6253
6254 \LWR@eqnarrayfactor
6255
6256 }

```

The starred version is patched to turn off the numbering:

```

6257 \csgpreto{eqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

The following $\mathcal{A}\mathcal{M}\mathcal{S}$ environments are more easily patched in-place:

Env `multline`

```

6258 \BeforeBeginEnvironment{multline}{
6259
6260 \ifbool{mathjax}
6261 {
6262   \LWR@syncmathjax
6263   \booltrue{LWR@amsmultline}
6264   \LWR@beginhideamsmath
6265 }
6266 {
6267   \lateximage[LWRAMSMATHBODY]
6268 }
6269 }
6270
6271 \AfterEndEnvironment{multline}{
6272
6273 \ifbool{mathjax}
6274 {
6275   \LWR@endhideamsmath
6276   \boolfalse{LWR@amsmultline}
6277   \LWR@addmathjax{multline}{\the\@envbody}
6278 }
6279 {\endlateximage}
6280
6281 }

```

Env `multline*`

```

6282 \BeforeBeginEnvironment{multline*}{
6283
6284 \ifbool{mathjax}
6285 {
6286   \LWR@syncmathjax

```

```
6287 \booltrue{LWR@amsmultline}
6288 \LWR@beginhideamsmath
6289 }
6290 {
6291 \lateximage[LWRAMSMATHBODY]
6292 }
6293 }
6294
6295 \AfterEndEnvironment{multline*}{
6296
6297 \ifbool{mathjax}
6298 {
6299 \LWR@endhideamsmath
6300 \boolfalse{LWR@amsmultline}
6301 \LWR@addmathjax{multline*}{\the\@envbody}
6302 }
6303 {\endlateximage}
6304
6305 }
6306
```

Env **gather**

```
6307 \BeforeBeginEnvironment{gather}{
6308
6309 \ifbool{mathjax}
6310 {
6311 \LWR@syncmathjax
6312 \boolfalse{LWR@amsmultline}
6313 \LWR@beginhideamsmath
6314 }
6315 {
6316 \lateximage[LWRAMSMATHBODY]
6317 }
6318 }
6319
6320 \AfterEndEnvironment{gather}{
6321
6322 \ifbool{mathjax}
6323 {
6324 \LWR@endhideamsmath
6325 \LWR@addmathjax{gather}{\the\@envbody}
6326 }
6327 {\endlateximage}
6328
6329 }
```

Env **gather***

```
6330 \BeforeBeginEnvironment{gather*}{
6331
6332 \ifbool{mathjax}
6333 {
6334   \LWR@syncmathjax
6335   \boolfalse{LWR@amsmultline}
6336   \LWR@beginhideamsmath
6337 }
6338 {
6339   \lateximage[LWRAMSMATHBODY]
6340 }
6341 }
6342
6343 \AfterEndEnvironment{gather*}{
6344
6345 \ifbool{mathjax}
6346 {
6347   \LWR@endhideamsmath
6348   \LWR@addmathjax{gather*}{\the\@envbody}
6349 }
6350 {\endlateximage}
6351
6352 }
```

Env **align**

```
6353 \BeforeBeginEnvironment{align}{
6354
6355 \ifbool{mathjax}
6356 {
6357   \LWR@syncmathjax
6358   \boolfalse{LWR@amsmultline}
6359   \LWR@beginhideamsmath
6360 }
6361 {
6362   \lateximage[LWRAMSMATHBODY]
6363 }
6364 }
6365
6366 \AfterEndEnvironment{align}{
6367
6368 \ifbool{mathjax}
6369 {
6370   \LWR@endhideamsmath
6371   \LWR@addmathjax{align}{\the\@envbody}
6372 }
6373 {\endlateximage}
```

```
6374
6375 }
```

Env `align*`

```
6376 \BeforeBeginEnvironment{align*}{
6377
6378 \ifbool{mathjax}
6379 {
6380   \LWR@syncmathjax
6381   \boolfalse{LWR@amsmultline}
6382   \LWR@beginhideamsmath
6383 }
6384 {
6385   \lateximage[LWRAMSMATHBODY]
6386 }
6387 }
6388
6389 \AfterEndEnvironment{align*}{
6390
6391 \ifbool{mathjax}
6392 {
6393   \LWR@endhideamsmath
6394   \LWR@addmathjax{align*}{\the\@envbody}
6395 }
6396 {\endlateximage}
6397
6398 }
```

Env `flalign`

```
6399 \BeforeBeginEnvironment{flalign}{
6400
6401 \ifbool{mathjax}
6402 {
6403   \LWR@syncmathjax
6404   \boolfalse{LWR@amsmultline}
6405   \LWR@beginhideamsmath
6406 }
6407 {
6408   \lateximage[LWRAMSMATHBODY]
6409 }
6410 }
6411
6412 \AfterEndEnvironment{flalign}{
6413
6414 \ifbool{mathjax}
6415 {
```

```

6416 \LWR@endhideamsmath
6417 \LWR@addmathjax{flalign}{\the\@envbody}
6418 }
6419 {\endlateximage}
6420
6421 }

```

Env `flalign*`

```

6422 \BeforeBeginEnvironment{flalign*}{
6423
6424 \ifbool{mathjax}
6425 {
6426 \LWR@syncmathjax
6427 \boolfalse{LWR@amsmultline}
6428 \LWR@beginhideamsmath
6429 }
6430 {
6431 \lateximage[LWRAMSMATHBODY]
6432 }
6433 }
6434
6435 \AfterEndEnvironment{flalign*}{
6436
6437 \ifbool{mathjax}
6438 {
6439 \LWR@endhideamsmath
6440 \LWR@addmathjax{flalign*}{\the\@envbody}
6441 }
6442 {\endlateximage}
6443
6444 }

6445 \end{warpHTML}

```

64 Lateximages

A `\lateximage` is typeset on its own PDF page inside an HTML comment which starts on the preceding page and ends on following page, and instructions are written to `lateximage.txt` for `lwarpmk` to extract the `\lateximage` from the page of the PDF file then generate an accompanying `.svg` file image file. Meanwhile, instructions to show this image are placed into the HTML file after the comment.

An HTML span is created to hold both the HTML comment, which will have the `pdftotext` conversion, and also the link to the final `.svg` image.

A \LaTeX label is used to remember which PDF page has the image. A label is used because footnotes, endnotes, and pagenotes may cause the image to appear at a later time. The label is declared along with the image, and so it correctly remembers where the image finally ended up.

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

for HTML output: 6446 `\begin{warpHTML}`

Ctrl `LWR@lateximagenumber` Sequence the images.

```
6447 \newcounter{LWR@lateximagenumber}
6448 \setcounter{LWR@lateximagenumber}{0}
```

Ctrl `LWR@lateximagedepth` Do not create `\lateximage` inside of `\lateximage`.

```
6449 \newcounter{LWR@lateximagedepth}
6450 \setcounter{LWR@lateximagedepth}{0}
```

A few utility macros to write special characters:

```
6451 \edef\LWR@hashmark{\string#} % for use in \write
6452 \edef\LWR@percent{\@percentchar} % for use in \write
```

Ctrl `LWR@LIpage` Used to reference the PDF page number of a lateximage to be written into `lateximages.txt`.

```
6453 \newcounter{LWR@LIpage}
```

```
6454 \end{warpHTML}
```

for HTML & PRINT: 6455 `\begin{warpall}`

`\LateximageFontSizeName` Declares how large to write text in the `\lateximage`. The `.svg` file text size should blend well with the surrounding HTML text size.

Do not include the leading backslash in the name.

```
6456 \newcommand*{\LateximageFontSizeName}{large}
```

```
6457 \end{warpall}
```

for HTML output: 6458 `\begin{warpHTML}`

`\LWR@HTMLsanitize` `{\text}`

Math expressions are converted to lateximages, and some math environments may contain “&”, “<”, or “>”, which should not be allowed inside an HTML ALT tag, so must convert them to HTML entities.

Two versions follow, depending on expansion needs. There may be a better way...

```
6459 \newcommand{\LWR@HTMLsanitize}[1]{%
6460 \begingroup%
6461 \LWR@FBcancel%
6462 \protect\StrSubstitute{\detokenize{#1}}%
6463 {\detokenize{&}}%
6464 {\detokenize{&}}[\LWR@strresult]%
6465 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6466 {\detokenize{<}}%
6467 {\detokenize{&lt;}}%
6468 [\LWR@strresult]%
6469 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6470 {\detokenize{>}}%
6471 {\detokenize{&gt;}}%
6472 [\LWR@strresult]%
6473 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6474 {\detokenize{##}}%
6475 {\#}%
6476 [\LWR@strresult]%
6477 \LWR@strresult%
6478 \endgroup%
6479 }
```

`\LWR@HTMLsanitizeexpand` $\{ \langle text \rangle \}$

This version expands the argument before sanitizing it.

```
6480 \newcommand{\LWR@HTMLsanitizeexpand}[1]{%
6481 \begingroup%
6482 \LWR@FBcancel%
6483 \protect\StrSubstitute{\detokenize\expandafter{#1}}%
6484 {\detokenize{&}}%
6485 {\detokenize{&}}[\LWR@strresult]%
6486 [\LWR@strresult]%
6487 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6488 {\detokenize{<}}%
6489 {\detokenize{&lt;}}%
6490 [\LWR@strresult]%
6491 \protect\StrSubstitute{\detokenize\expandafter{\LWR@strresult}}%
6492 {\detokenize{>}}%
6493 {\detokenize{&gt;}}%
6494 [\LWR@strresult]%
6495 \LWR@strresult%
6496 \endgroup%
```

6497 }

Env lateximage [*alttag*]

```
6498 \catcode'\$=\active%
6499
6500 \NewDocumentEnvironment{lateximage}{0{image}}
6501 {%
6502 \LWR@traceinfo{lateximage: starting on page \arabic{page}}%
6503 \ifthenelse{\cnttest{\value{LWR@lateximagedepth}}{>}{0}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6504 {%
6505   \addtocounter{LWR@lateximagedepth}{1}%
6506 }%
```

Otherwise, this is the outer-most lateximage:

```
6507 {% start of outer-most lateximage
```

Starting a new lateximage:

```
6508   \addtocounter{LWR@lateximagenumber}{1}%
6509   \LWR@traceinfo{lateximage: LWR@lateximagenumber is \arabic{LWR@lateximagenumber}}%
```

While inside a lateximage, locally do not use mathjax:

```
6510   \boolfalse{mathjax}
```

Be sure that are doing a paragraph:

```
6511   \LWR@ensuredoingapar%
```

Next file:

```
6512   \addtocounter{LWR@externalfilecnt}{1}%
6513   \LWR@traceinfo{lateximage: LWR@externalfilecnt is \arabic{LWR@externalfilecnt}}%
```

Figure out what the next page number will be:

```
6514   \setcounter{pageref}{LWR@LIpage}\theLWR@lateximagenumber}%
6515   \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:

```
6516 \LWR@htmltag{span id="lateximage\arabic{LWR@lateximagenumber}" %
6517 class="lateximagesource"{}} \LWR@orignewline
```

Write instructions to the `lateximages.txt` file:

```
6518 \LWR@traceinfo{lateximage: about to write to lateximages.txt}%
6519 \immediate\write\LWR@lateximagesfile{|\theLWR@LPage|\theLWR@externalfilecnt|}%
```

Place an open comment tag at the bottom of page; footnotes will be above this tag. This will hide any traces of the lateximage PDF page which were picked up by `pdftotext`.

```
6520 \LWR@traceinfo{lateximage: about to create open comment}%
6521 \LWR@htmlopencomment%
```

One level deeper:

```
6522 \addtocounter{LWR@lateximagedepth}{1}%
```

Start the new PDF page:

```
6523 \LWR@traceinfo{lateximage: about to create new page}%
6524 \LWR@orignewpage%
```

Typeset the image in a “standard” width page and font size:

```
6525 \LWR@traceinfo{lateximage: about to create minipage}%
6526 \LWR@origminipage{6in}%
6527 \csuse{LWR@orig\LateximageFontSizeName}%
```

Temporarily restore formatting to its PDF definitions: Do not produce HTML tags for `\hspace`, etc. inside a lateximage.

```
6528 \LWR@traceinfo{lateximage: about to temporarily restore formatting}%
6529 \LWR@restoreorigformatting%
```

Use full-page footnotes instead of minipage footnotes. These become HTML footnotes.

```
6530 \def\@mpfn{footnote}%
6531 \def\thempfn{\thefootnote}%
6532 \let\@footnotetext\LWR@footnotetext%
```

Create the `LWRlateximage<number>` label:

```
6533 \LWR@traceinfo{lateximage: about to create label}%
6534 \LWR@origlabel{LWRlateximage\arabic{LWR@lateximagenumber}}%
6535 \LWR@traceinfo{lateximage: finished creating the label}%
```

Enable print-mode math functions:

```
6536 \LetLtxMacro$\LWR@origdollar%
6537 \catcode'\$=3% math shift
6538 \LetLtxMacro\(\LWR@origopenparen%
6539 \LetLtxMacro\)\LWR@origcloseparen%
6540 \LetLtxMacro\[ \LWR@origopenbracket%
6541 \LetLtxMacro\] \LWR@origclosebracket%
6542 \let\@ensuredmath\LWR@orig@ensuredmath%
6543 }% end of outer-most lateximage
6544 \LWR@traceinfo{lateximage: finished start of environment}%
6545 }% end of \begin{lateximage}
```

When the environment closes:

```
6546 {% start of \end{lateximage}
6547 \LWR@traceinfo{lateximage: starting end of environment}%
```

Nested more than one deep?

```
6548 \ifthenelse{\cinttest{\value{LWR@lateximagedepth}}{>}{1}}%
```

If nesting inside an already-existing lateximage, simply record one more level:

```
6549 {\addtocounter{LWR@lateximagedepth}{-1}}%
```

If this is the outer-most lateximage:

```
6550 {% end of outer-most lateximage
```

Finish the lateximage minipage and start a new PDF page:

```
6551 \LWR@origendminipage%
6552 \LWR@orignewpage%
6553 \LWR@origscriptsize%
```

Close the HTML comment which encapsulated any traces of the lateximage picked up by pdftotext:

```
6554 \LWR@htmlclosecomment{\LWR@orignewline%
6555 \LWR@traceinfo{lateximage: The page after the image is \arabic{page}}%
```

Create a link to the lateximage, allowing its natural height:

If the alt tag is given as “LWRAMSMATHBODY”, then use the text collected by the amsmath multiline, gather, or align environments.

```
6556 \ifthenelse{\equal{#1}{LWRAMSMATHBODY}}%
```

```

6557   {%
6558   \LWR@subinlineimage[%
6559   \LWR@HTMLsanitizeexpand{\detokenize\expandafter{\the\@envbody}}}%
6560   ]%
6561   {lateximage}%
6562   {lateximages\OSPathSymbol{}}lateximage-\theLWR@externalfilecnt}%
6563   {svg}%
6564   }{}%
6565   }%
6566   {%
6567   \LWR@subinlineimage[#1]{lateximage}%
6568   {lateximages\OSPathSymbol{}}lateximage-\theLWR@externalfilecnt}{svg}{}%
6569   }%
6570 % \LWR@orignewline% Removed to prevent extra space.

```

Be sure that are doing a paragraph:

```
6571   \LWR@ensuredoingapar%
```

Close the HTML span which has the pdftotext comment and also the link to the .svg image:

```

6572   \LWR@htmltag{/span}%
6573   \ifbool{HTMLDebugComments}{%
6574     \LWR@htmlcomment{End of lateximage}%
6575   }{}%
6576 % \LWR@orignewline% Removed to prevent extra space.

```

Undo one lateximage level:

```

6577   \addtocounter{LWR@lateximagedepth}{-1}%
6578 }% end of outer-most lateximage
6579 \LWR@traceinfo{lateximage: done}%
6580 }%
6581 \catcode'\$=3% math shift
6582 \end{warpHTML}

```

```

for PRINT output: 6583 \begin{warpprint}
6584 % \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6585 \newenvironment{lateximage}[1][\minipage{\linewidth}]{\endminipage}
6586 \end{warpprint}

```

65 center, flushleft, flushright

```
for HTML output: 6587 \begin{warpHTML}
```

Env `center` Replace `center` functionality with CSS tags:

```
6588 \renewenvironment*{center}
6589 {
6590 \LWR@forcenewpage
6591 \BlockClass{center}
6592 }
6593 {\endBlockClass}
```

Env `flushright`

```
6594 \renewenvironment*{flushright}
6595 {
6596 \LWR@forcenewpage
6597 \BlockClass{flushright}
6598 }
6599 {\endBlockClass}
```

Env `flushleft`

```
6600 \renewenvironment*{flushleft}
6601 {
6602 \LWR@forcenewpage
6603 \BlockClass{flushleft}
6604 }
6605 {\endBlockClass}

6606 \end{warpHTML}
```

66 Pre-loaded packages

for HTML output: 6607 \begin{warpHTML}

If `textcomp` was loaded before `lwarp`, perhaps as part of the font-related packages, explicitly load the `lwarp` patches now:

```
6608 \@ifpackageloaded{textcomp}
6609 {
6610 \LWR@origRequirePackage{lwarp-textcomp}
6611 }
6612 {}
```

If `graphics` or `graphicx` were loaded before `lwarp`, perhaps by `xunicode`, explicitly load the `lwarp` patches now:

```

6613 \@ifpackageloaded{graphics}
6614 {
6615 \LWR@origRequirePackage{lwarp-graphics}
6616 }
6617 {}

6618 \end{warpHTML}

```

67 Siunitx

Pkg siunitx



per-mode

Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

Some units will require that the expression be placed inside math mode.

for HTML output: 6619 `\begin{warpHTML}`

Options for `siunitx`:

```

6620 \PassOptionsToPackage{
6621     detect-mode=true,
6622     per-mode=symbol,% fraction is not seen by pdftotext
6623 %     text-celsius = {\protect\LWRsiunitx@degree{ }C},
6624 %     text-degree = {\protect\LWRsiunitx@degree},
6625 }{siunitx}
6626

6627 \end{warpHTML}

```

68 Graphics

Pkg graphics

Pkg graphicx



.pdf image files

For `\includegraphics` with `.pdf` files, the user should provide a `.pdf` image file, and also a `.svg`, `.png`, or `.jpg` version of the same image. **These should be referred**



no file extension

to without a file extension:

```
\includegraphics{filename} % print:.pdf, HTML:.svg or other
```

For print output, lwarp will automatically choose the .pdf if available, other some other format otherwise. For HTML, one of the other formats is used instead.

If a .pdf file is explicitly referred to with its file extension, a link to the .pdf file will appear in the HTML output.

```
\includegraphics{filename.pdf} % creates a link in HTML
```

other image files For .png, .jpg, or .gif image files, the same file may be used in both print or HTML versions, and may be used with a file extension, but will also be used without the file extension if it is the only file of its base name.

⚠ **graphics vs. graphicx** If using the older graphics syntax, use both optional arguments for `\includegraphics`. A single optional parameter is interpreted as the newer graphicx syntax. Note that

⚠ **viewports** viewports are not supported by warp; the entire image will be shown.

units For `\includegraphics`, avoid px and % units for width and height, or enclose them inside `warpHTML` environments. For font-proportional image sizes, use ex or em. For fixed-sized images, use cm, mm, in, pt, or pc. Use the keys `width=.5\linewidth`, or similar for `\textwidth` or `\textheight` to give fixed-sized images proportional to a 6 by 9 inch text area.

options `\includegraphics` accepts width and height, origin, rotate and scale, plus a new class key.

HTML class With HTML output, `\includegraphics` accepts an optional `class=xyz` keyval combination, and if this is given then the HTML output will include that class for the image. The class is ignored for print output.

`\rotatebox` `\rotatebox` accepts the optional origin key.

⚠ **browser support** `\rotatebox`, `\scalebox`, and `\reflectbox` depend on modern browser support. The CSS standard declares that when an object is transformed the whitespace which they occupied is preserved, unlike \TeX , so expect some ugly results for scaling and rotating.

for PRINT output: For print output, accept and then discard the new class key:

```
6628 \begin{warpprint}
6629 \define@key{Gin}{class}{}
6630 \end{warpprint}
```

68.1 xcolor print-mode additions

Print-mode versions of new xcolor definitions. These are defined inside `warpall` because they are also used for HTML while inside a `lateximage`. They are defined

`\AtBeginDocument` so that the `xcolor` originals may first be loaded and saved for reuse.

The framed versions are modified to allow a background color of none, in which case only the frame is drawn, allowing the background page color to show.

for HTML & PRINT: 6631 `\begin{warppall}`

After xparse may have been loaded ...

6632 `\AtBeginDocument{`

... and only if xcolor was loaded:

6633 `\@ifpackageloaded{xcolor}{`

6634 `\LWR@traceinfo{patching xcolor}`

`\colorboxBlock` `\colorboxBlock` is the same as `\colorbox`:

6635 `\LetLtxMacro\colorboxBlock\colorbox`

In HTML mode, the following is done when `xcolor` is loaded. Following is the print-mode action:

6636 `\warpprintonly{`

6637 `\LetLtxMacro\LWRprint@colorboxBlock\colorbox`

6638 `\LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox`

6639 `\LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox`

6640 `}`

`\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`.)

6641 `\newsavebox{\LWR@colorminipagebox}`

6642

6643 `\DeclareDocumentCommand{\LWRprint@fcolorbox}{o m o m +m}{%`

6644 `\LWR@traceinfo{\LWRprint@fcolorbox #2 #4}%`

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

6645 `\begin{lrbox}{\LWR@colorminipagebox}%`

6646 `#5%`

6647 `\end{lrbox}%`

Sort out the various optional arguments and the background color of none. In each case, the LRbox is placed inside a `\fcolorbox`.

```

6648 \ifthenelse{\equal{#4}{none}}%
6649 {% #4 none
6650   \LWR@traceinfo{background is none}%
6651   {% scope the \colorlet
6652     \colorlet{LWR@currentcolor}{.}%
6653     \color{#2}%
6654     \fbox{%
6655       \color{LWR@currentcolor}%
6656       \usebox{\LWR@colorminipagebox}%
6657     }% fbox
6658   }% colorlet
6659 }% #4 none
6660 {% #4 not none
6661 \LWR@traceinfo{background not none}%
6662 \IfValueTF{#1}%
6663 {%
6664   \IfValueTF{#3}%
6665   {\LWR@origprint@fcolorbox[#1]{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6666   {\LWR@origprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6667 }%
6668 {% no value #1
6669   \IfValueTF{#3}%
6670   {\LWR@origprint@fcolorbox{#2}{#3}{#4}{\usebox{\LWR@colorminipagebox}}}%
6671   {\LWR@origprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6672 }% no value #1
6673 }% #4 not none
6674 \LWR@traceinfo{LWRprint@fcolorbox done}%
6675 }

```

`\fcolorboxBlock` [*framemodel*] [*framecolor*] [*boxmodel*] [*boxcolor*] [*text*]

In print mode, `\fcolorboxBlock` is the same as `\fcolorbox`.

```

6676 \LetLtxMacro\LWRprint@fcolorboxBlock\LWRprint@fcolorbox

```

Env `fcolorminipage` [*1:framemodel*] [*2:framecolor*] [*3:boxmodel*] [*4:boxcolor*] [*5:align*] [*6:height*] [*7:inner-align*] [*8:width*]

In print mode, becomes a `\fcolorbox` containing a minipage:

```

6677 \NewDocumentEnvironment{LWRprint@fcolorminipage}{o m o m O{c} O{ } o m}
6678 {%
6679 \LWR@traceinfo{*** fcolorminipage: #2 #4 #8}%

```

Pre-load the contents into an LR box so that they can be used inside a `\fcolorbox`:

```
6680 \begin{lrbox}{\LWR@colorminipagebox}%
```

If inner alignment is not given, use the outer alignment instead:

```
6681 \IfValueTF{#7}%
6682 {\begin{minipage}[#5][#6][#7]{#8}}%
6683 {\begin{minipage}[#5][#6][#5]{#8}}%
6684 }%
6685 {%
6686 \end{minipage}%
6687 \end{lrbox}%
6688 \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`.

```
6689 \ifthenelse{\equal{#4}{none}}%
6690 {% #4 none
6691   {% scope the \colorlet
6692     \colorlet{\LWR@currentcolor}{.}%
6693     \color{#2}%
6694     \fbox{%
6695       \color{\LWR@currentcolor}%
6696       \usebox{\LWR@colorminipagebox}%
6697     }% fbox
6698   }% colorlet
6699 }% #4 none
6700 {% #4 not none
6701   \IfValueTF{#1}%
6702   {%
6703     \IfValueTF{#3}%
6704     {\LWRorigprint@fcolorbox[#1]{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
6705     {\LWRorigprint@fcolorbox[#1]{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6706     }%
6707     {% no value #1
6708     \IfValueTF{#3}%
6709     {\LWRorigprint@fcolorbox{#2}[#3]{#4}{\usebox{\LWR@colorminipagebox}}}%
6710     {\LWRorigprint@fcolorbox{#2}{#4}{\usebox{\LWR@colorminipagebox}}}%
6711     }% no value #1
6712 }% #4 not none
6713 \LWR@traceinfo{*** finished end fcolorminipage}%
6714 }
```

`\LWR@restoreorigprintxcolor` Used to activate print-mode additions for xcolor. In print mode, this is used immediately following. In HTML mode, this is used inside a `lateximage`.

```

6715 \newcommand*{\LWR@restoreorigprintxcolor}{%
6716 \LWR@traceinfo{\LWR@restoreorigprintxcolor}%
6717 \LetLtxMacro\colorboxBlock\LWRprint@colorboxBlock%
6718 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
6719 \LetLtxMacro\fcolorboxBlock\LWRprint@fcolorboxBlock%
6720 \LetLtxMacro\fcolorminipage\LWRprint@fcolorminipage%
6721 \LetLtxMacro\endfcolorminipage\endLWRprint@fcolorminipage%
6722 }
6723
6724 \appto{\LWR@restoreorigformatting}{%
6725 \LWR@restoreorigprintxcolor%
6726 }

```

If print mode, immediately activate the print-mode enhancements for xcolor:

```

6727 \warpprintonly{\LWR@restoreorigprintxcolor}
6728
6729 \LWR@traceinfo{xcolor patches done}
6730 }{}% xcolor loaded
6731 }% AtBeginDocument

6732 \end{warpall}

```

69 Cleveref

Pkg `cleveref` cleveref package is used as-is with minor patches.

 **cleveref page numbers** cleveref and varioref are supported, but printed page numbers do not map to HTML, so a section name or a text phrase are used for `\cpageref` and `\cpagerefrange`. This phrase includes `\cpagerefFor`, which defaults to “for”.

Ex:

```

\cpageref{tab:first,tab:second}
in HTML becomes:
“pages for table 4.1 and for table 4.2”

```

See `\cpagerefFor` at page 357 to redefine the message which is printed for page number references.

loading order cleveref and the following associated macro patches are automatically preloaded at the end of the preamble via `\AtEndPreamble` and `\AfterEndPreamble`. This is done because the HTML conversion requires cleveref. The user’s document may not require cleveref, thus the user may never explicitly load it, so during HTML output lwarp loads it last. If the user’s document preamble uses cleveref options, or functions such

as `\crefname`, then `cleveref` may be loaded in the user's preamble near the end, and `lwarp`'s additional loading of `cleveref` will have no effect.

Table 8 on page 301 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: 6733 `\begin{warpHTML}`

`\AtEndPreamble` forces `cleveref` to be loaded last:

```
6734 \AtEndPreamble{
6735 \RequirePackage{cleveref}
6736 }
```

The following patches are applied after `cleveref` has loaded, and after `\AtBeginDocument`:

```
6737 \AfterEndPreamble{
6738 \LWR@traceinfo{Patching cleveref.}
```

`\@@setcref` `{\langle kindofref \rangle}{\langle label \rangle}`

```
6739 \renewcommand*\@@setcref}[2]{#1{\ref{#2}}{-}{}}
```

`\@@setcrefrange` `{\langle text \rangle}{\langle label \rangle}{\langle label \rangle}`

```
6740 \renewcommand{\@@setcrefrange}[3]{%
6741 #1{\ref{#2}}{\ref{#3}}{-}{-}{}}
```

`\cpagerefFor` Redefinable word between “page(s)” and the page numbers.

```
6742 \newcommand*\cpagerefFor{for}
```

`\@@setcpageref` `{\langle typeofref \rangle}{\langle label \rangle}`, where `typeofref` is “page” or “pages”

```
6743 \renewcommand*\@@setcpageref}[2]{%
6744 #1{\cpagerefFor\ \cref{#2}}{-}{}%
6745 }
```

```
6746 \renewcommand{\@@setcpagerefrange}[3]{%
6747 #1{\cpagerefFor\ \cref{#2}}{\cref{#3}}{-}{-}{}}
6748 }% AfterEndPreamble
```

Remember and patch some label-related definitions. These will be further encased and patched by other packages later.

```
6749 \let\LWR@origlabel\label
6750 \let\label\LWR@newlabel
6751 \let\LWR@origref\ref
6752 \let\ref\LWR@newref% \end{ syntax highlighting
6753 \let\LWR@origpageref\pageref
6754 \let\pageref\LWR@newpageref
6755
6756
6757
6758 \end{warpHTML}
```

70 Picture

Env `picture` The `picture` environment is enclosed inside a `\lateximage`.

for HTML output: 6759 `\begin{warpHTML}`

Env `picture`

```
6760 \BeforeBeginEnvironment{picture}{\lateximage}
6761
6762 \AfterEndEnvironment{picture}{\endlateximage}

6763 \end{warpHTML}
```

71 Boxes and Minipages

A CSS flexbox is used for minipages and parboxes, allowing external and internal vertical positioning.

 **inline** A line of text with an inline minipage or parbox will have the minipage or parbox placed onto its own line, because a paragraph is a block element and cannot be made inline-block.

placement Minipages and parboxes will be placed side-by-side in HTML unless you place a `\newline` between them.

side-by-side Side-by-side minipages may be separated by `\quad`, `\qquad`, `\enskip`, `\hspace`, `\hfill`, or a `\rule`. When inside a `center` environment, the result is similar in print

and HTML. Paragraph tags are suppressed between side-by-side minipages and these spacing commands, but not at the start or end of the paragraph.

in a span There is limited support for minipages inside an HTML ``. An HTML `<div>` cannot appear inside a ``. While in a ``, minipages, and parboxes, and any enclosed lists have limited HTML tags, resulting in an “inline” format, without markup except for HTML breaks. Use `\newline` or `\par` for an HTML break.

size When using `\linewidth`, `\textwidth`, and `\textheight`, widths and heights are scaled proportionally to a 6×9 inch text area.

no-width minipages A minipage of width exactly `\linewidth` is automatically given no HTML width.

full-width minipages A new macro `\minipagefullwidth` requests that the next minipage be generated without an HTML width attribute, allowing it to be the full width of the display rather than the fixed width given.

 **text alignment** Nested minipages adopt their parent’s text alignment in HTML, whereas in regular \TeX PDF output they do not. Use a `flushleft` or similar environment in the child minipage to force a text alignment.

for HTML output: 6764 `\begin{warpHTML}`

71.1 Counters and lengths

Ctr `LWR@minipagedepth` Used to only reset the line width at the outermost minipage.

```
6765 \newcounter{LWR@minipagedepth}
6766 \setcounter{LWR@minipagedepth}{0}
```

Len `\WR@minipagewidth` Used to convert the width into printable units.

```
6767 \newlength{\LWR@minipagewidth}
```

Len `\WR@minipageheight` Used to convert the height into printable units.

```
6768 \newlength{\LWR@minipageheight}
```

71.2 Footnote handling

Also see section 45 for other forms of footnotes. Minipage footnotes are gathered in section 45.5, and then placed into the document in section 71.3.

71.3 Minipage handling

`\LWR@endminipage` Used to close a minipage.

Copied the \TeX definition and modified to create a `<div>` of class `mpfootnotes`:

```
6769 \def\LWR@endminipage{%
6770   \par
6771   \unskip
6772   \ifvoid\@mpfootins\else
6773     \vskip\skip\@mpfootins
6774     \normalcolor
6775     \LWR@htmldivclass{mpfootnotes}
6776     \LWR@origmedskip
6777     \unvbox\@mpfootins
6778     \LWR@htmldivclassend{mpfootnotes}
6779   \fi
6780   \@minipagefalse
6781   \color@endgroup
6782   \egroup
6783   \expandafter\@iiparbox\@mpargs{\unvbox\@tempboxa}}
```

`\LWR@subminipage` Used to create a PDF minipage without creating an HTML minipage. This allows footnotes to appear at the bottom of the minipage instead of the bottom of the HTML page.

```
6784 \newcommand*\LWR@subminipage}{%
6785 \LWR@stoppars
6786 \LWR@origminipage{6in}
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6787 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6788 \LWR@startpars%
6789 }
```

`\LWR@endsubminipage` Closes the subminipage.

```
6790 \newcommand*\LWR@endsubminipage}{%
6791 \LWR@stoppars%
6792 \LWR@endminipage% The following empty line is required:
6793
6794 }
```

Bool LWR@minipagefullwidth Should the next minipage have no HTML width?

```

6795 \newbool{LWR@minipagefullwidth}
6796 \boolfalse{LWR@minipagefullwidth}

```

`\minipagefullwidth` Requests that the next minipage have no width tag in HTML:

for HTML output:

```

6797 \newcommand*{\minipagefullwidth}{\booltrue{LWR@minipagefullwidth}}
6798 \end{warpHTML}

```

for PRINT output:

```

6799 \begin{warpprint}
6800 \newcommand*{\minipagefullwidth}{}
6801 \end{warpprint}

```

for HTML output:

```

6802 \begin{warpHTML}

```

Bool LWR@minipagethispar Has a minipage been seen this paragraph? If true, prevents paragraph tags around horizontal space between minipages.

```

6803 \newbool{LWR@minipagethispar}
6804 \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.

```

6805 \RenewDocumentEnvironment{minipage}{0{t} o 0{t} m}
6806 {%
6807 \uselengthunit{in}%
6808 \setlength{\LWR@minipagewidth}{#4}%
6809 \ifthenelse{\cinttest{\value{LWR@minipagedepth}}{=}{0}}{%
6810   \addtolength{\LWR@minipagewidth}{3em}% room for frames
6811   \setlength{\linewidth}{6in}%
6812   \setlength{\textwidth}{6in}%
6813   \setlength{\textheight}{9in}%
6814 }{}%
6815 \LWR@traceinfo{computed width is \rndprintlength{\LWR@minipagewidth}}
6816 \addtocounter{LWR@minipagedepth}{1}%
6817 \setlength{\LWR@minipageheight}{\textheight}% default unless specified
6818 \IfValueT{#2}{\setlength{\LWR@minipageheight}{#2}}%

```

ℒ_{TEX} wants to start a paragraph for the new minipage, then start a paragraph again for the contents of the minipage, so cancel the paragraph tag handling until the minipage has begun.

6819 \LWR@stoppars%

Create the <div> tag with optional alignment style:

```
6820 \LWR@traceinfo{minipage: creating div class}%
6821 \LWR@orignewpage%
6822 \LWR@htmltag{div class="minipage" style="%
6823 \ifthenelse{\equal{#1}{t}}{vertical-align:bottom ; }{}%
6824 \ifthenelse{\equal{#1}{c}}{vertical-align:middle ; }{}%
6825 \ifthenelse{\equal{#1}{b}}{vertical-align:top ; }{}%
6826 \ifthenelse{\equal{#3}{t}}{justify-content:flex-start ; }{}%
6827 \ifthenelse{\equal{#3}{c}}{justify-content:center ; }{}%
6828 \ifthenelse{\equal{#3}{b}}{justify-content:flex-end ; }{}%
6829 \ifthenelse{\equal{#3}{s}}{justify-content:space-between ; }{}%
```

Print the width and optional height styles:

```
6830 \LWR@traceinfo{minipage: about to print the width of \rndprintlength{\LWR@minipagewidth}}%
6831 \uselengthunit{PT}%
6832 \ifbool{LWR@minipagefullwidth}%
6833 {\boolfalse{LWR@minipagefullwidth}}%
6834 {%
6835   \ifthenelse{\lengthtest{#4}=\linewidth}%
6836   {}%
6837   {width:\rndprintlength{\LWR@minipagewidth} ; }%
6838 }%
6839 \LWR@traceinfo{minipage: about to print the height}%
6840 \IfValueT{#2}{height:\rndprintlength{\LWR@minipageheight} ; }%
6841 "{}%
```

Finish with an empty line to start \LaTeX minipage processing on a new line. Use a large minipage area to avoid the unnecessary wrapping of tags.

```
6842
6843 \LWR@origminipage{6in}% The preceding empty line is required.
```

Set the user-accessible minipage and text width and height values inside the minipage. These do not affect the actual size of the large minipage created by `\LWR@origminipage` above, but are used by any reference to `\linewidth`, etc. inside the PDF minipage being created here.

```
6844 \setlength{\linewidth}{#4}% the original width
6845 \setlength{\textwidth}{6in}%
6846 \setlength{\textheight}{9in}%
```

`\raggedright` cancels hyphenation, which will be done by HTML instead.

```
6847 \LWR@origraggedright%
```

Resume paragraph tag handling for the contents of the minipage:

```
6848 \LWR@startpars%
6849 \LWR@traceinfo{minipage: finished starting the minipage}%
6850 }
```

End the environment with \LaTeX processing and closing tag:

```
6851 {%
6852 \LWR@stoppars%
6853 \LWR@endminipage% The following empty line is required:
6854
6855 \LWR@htmldivclassend{minipage}%
6856 \LWR@origvspace{1\baselineskip}% required for subcaption
6857 \addtocounter{LWR@minipagedepth}{-1}%
6858 \LWR@startpars%
```

Prevent paragraph tags around horizontal white space until the start of the next paragraph:

```
6859 \global\booltrue{LWR@minipagethispar}%
6860 }
```

71.4 Parbox, makebox, framebox, fbox, raisebox

for HTML output:
 \backslash parbox [*<pos>*] [*<height>*] [*<inner-pos>*] [*<width>*] [*<text>*]

A parbox uses the minipage code:

```
6861 \RenewDocumentCommand{\parbox}{O{t} o O{t} m +m}
6862 {
6863 \LWR@traceinfo{parbox of width #4}%
6864 \begin{minipage}[#1][#2][#3]{#4}
6865 #5
6866 \end{minipage}
6867 }
```

\backslash makebox (<(>)posn) [*<width>*] [*<pos>*] [*<text>*]

```
6868 \LetLtxMacro\LWR@origmakebox\makebox
6869
6870 \RenewDocumentCommand{\makebox}{d() o o m}{%
```

Check for the optional width:

```
6871 \IfValueTF{#2}%
6872 {%
```

Check for the horizontal text alignment. For stretched, the best HTML can do is justified alignment.

```
6873   {% scope
6874   \def\LWR@align{center}%
6875   \ifstrequal{#3}{l}{\def\LWR@align{left}}{}%
6876   \ifstrequal{#3}{r}{\def\LWR@align{right}}{}%
6877   \ifstrequal{#3}{s}{\def\LWR@align{justify}}{}%
```

To print the width argument:

```
6878   \setlength{\LWR@tempwidth}{#2}%
```

inline-block allows width and text-alignment to be used in a .

```
6879   \uselengthunit{PT}%
6880   \InlineClass[%
6881     display:inline-block ; %
6882     text-align:\LWR@align\ ; %
6883     width:\rndprintlength{\LWR@tempwidth}%
6884   ]%
6885   {makebox}%
```

Without a width argument, the text is simply used inline:

```
6886   {\mbox{#4}}%
6887   }% scope
6888 }%
6889 {\mbox{#4}}%
6890 }
```

`\framebox` [*width*] [*pos*] {*text*}

```
6891 \LetLtxMacro\LWR@origframebox\framebox
6892
6893 \RenewDocumentCommand{\framebox}{o o m}{%
6894 \fbox{\makebox[#1][#2]{#3}}%
6895 }
```

`\LWR@forceminwidth` {*length*}

Sets `\LWR@tempwidth` to be at least 1pt.

```
6896 \newlength{\LWR@atleastonept}
```

```

6897
6898 \newcommand*\LWR@forceminwidth}[1]{%
6899 \setlength{\LWR@atleastonept}{#1}%
6900 \ifthenelse{%
6901   \lengthtest{\LWR@atleastonept>0pt}\AND%
6902   \lengthtest{\LWR@atleastonept<1pt}}%
6903 }%
6904 {\setlength{\LWR@atleastonept}{1pt}}%
6905 {}%
6906 }

```

`\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.

```

6907 \newcommand*\LWR@blackborderpadding){%
6908 \uselengthunit{PT}%
6909 border:\rndprintlength{\LWR@atleastonept} solid black ; %
6910 padding:\rndprintlength{\fboxsep}%
6911 }

```

`\fbox` $\{ \langle text \rangle \}$

Creates a framed inline span enclosing the text.

Remember the print-mode version:

```
6912 \let\LWRprint@fbox\fbox
```

Create a new HTML version, but don't use it until after `xcolor` may have loaded:

```

6913 \newcommand{\LWRhtml@fbox}[1]{%
6914 \LWR@traceinfo{HTML fbox}
6915 \LWR@forceminwidth{\fboxrule}%
6916 \InlineClass[%
6917 \LWR@blackborderpadding%
6918 ]{fbox}{#1}
6919 }

```

`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`:

```
6920 \AtBeginDocument{\let\fbox\LWRhtml@fbox}
```

`\fboxBlock` $\{ \langle text \rangle \}$ Creates a framed HTML `<div>` of the text.

A print-output version is also supplied below.

```

6921 \newcommand{\fboxBlock}[1]{%
6922 \LWR@forceminwidth{\fboxrule}%
6923 \begin{BlockClass}[%
6924 \LWR@blackborderpadding%
6925 ]{fboxBlock}
6926 #1
6927 \end{BlockClass}
6928 }

```

Env `fminipage` [*align*] [*height*] [*align*] [*width*]

Creates a framed HTML <div> around its contents.

A print-output version is also supplied below.

```

6929 \NewDocumentEnvironment{fminipage}{0{t} o 0{t} m}
6930 {%
6931 \LWR@traceinfo{fminipage #1 #2 #3 #4}%
6932 \LWR@forceminwidth{\fboxrule}%
6933 \setlength{\LWR@tempwidth}{#4}%
6934 \IfValueT{#2}{\setlength{\LWR@tempheight}{#2}}%
6935 \begin{BlockClass}[%
6936 \LWR@blackborderpadding ; %
6937 \uselengthunit{PT}%
6938 \IfValueT{#2}{height:\rndprintlength{\LWR@tempheight} ; }%
6939 width:\rndprintlength{\LWR@tempwidth}%
6940 ]{fminipage}%
6941 }
6942 {%
6943 \end{BlockClass}%
6944 \LWR@traceinfo{fminipage done}%
6945 }

```

`\raisebox` [*raiselen*] [*height*] [*depth*] [*text*]

```

6946 \LetLtxMacro{\LWR@origraisebox}{\raisebox}
6947
6948 \RenewDocumentCommand{\raisebox}{m o o m}{%
6949 #4%
6950 }
6951 \end{warpHTML}

```

for HTML & PRINT: 6952 \begin{warpall}

`LWRprint@fminipage` is defined inside `warpall`. For print output, it is `\let` to `fminipage`. For HTML output, the HTML version of `fminipage` is used instead, but the print version is still available for use inside a `lateximage`.

Env LWRprint@fminipage [*1:align*] [*2:height*] [*3:inner-align*] [*4:width*]

Creates a frame around its contents.

```
6953 \newsavebox{\LWR@fminipagebox}
6954
6955 \NewDocumentEnvironment{LWRprint@fminipage}{0{t} o 0{t} m}
6956 {%
```

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:

```
6957 \IfValueTF{#3}%
6958 {\def\LWR@thisalign{#3}}
6959 {\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.

```
6960 \IfValueTF{#2}%
6961 {\minipage[#1][#2+2\fboxsep+2\fboxrule][\LWR@thisalign]{#4+2\fboxsep+2\fboxrule}}%
6962 {\minipage[#1]{#4+2\fboxsep+2\fboxrule}}%
```

Capture the contents of the environment:

```
6963 \begin{lrbox}{\LWR@fminipagebox}%
```

Nest the contents inside an inner minipage of the desired size:

```
6964 \IfValueTF{#2}%
6965 {\minipage[#1][#2][\LWR@thisalign]{#4}}%
6966 {\minipage[#1]{#4}}%
6967 }
6968 {%
```

Close the inner minipage and the LR box with the contents:

```
6969 \endminipage%
6970 \end{lrbox}%
```

Create a frame around the contents of the environment:

```
6971 \fbox{\usebox{\LWR@fminipagebox}}%
```

The entire thing is placed inside the outer minipage:

```
6972 \endminipage%
6973 }
```

```
6974 \end{warppall}
```

for PRINT output: 6975 \begin{warpprint}

For print output, the following are \let to become active.

```
\fboxBlock {<text>}
```

Creates a framed HTML <div> around the text.

```
6976 \let\fboxBlock\fbox
```

```
Env fminipage [align] [height] [align] {<width>}
```

Creates a frame around its contents.

```
6977 \LetLtxMacro{\fminipage}{\LWRprint@fminipage}
6978 \LetLtxMacro{\endfminipage}{\endLWRprint@fminipage}
```

```
6979 \end{warpprint}
```

72 Direct formatting

\textbf, etc. are supported, but \bfseries, etc. are not yet supported.

For high-level block and inline custom CSS classes, see section 39.8.

for HTML output: 6980 \begin{warppHTML}

```
\emph {<text>}
```

```
6981 \DeclareRobustCommand{\LWR@HTMLemph}[1]{\LWR@htmlspan{em}{#1}}
6982 \DeclareRobustCommand{\LWR@nullemph}[1]{#1}
6983 \LetLtxMacro{\emph}{\LWR@HTMLemph}
```

```
\textmd {<text>}
```

```
6984 \DeclareRobustCommand{\LWR@HTMLtextmd}[1]{\LWR@htmlspanclass{textmd}{#1}}
6985 \DeclareRobustCommand{\LWR@nulltextmd}[1]{#1}
6986 \LetLtxMacro{\textmd}{\LWR@HTMLtextmd}
```

`\textbf` $\langle text \rangle$

```
6987 \DeclareRobustCommand{\LWR@HTMLtextbf}[1]{\LWR@htmlspan{b}{#1}}
6988 \DeclareRobustCommand{\LWR@nulltextbf}[1]{#1}
6989 \LetLtxMacro{\textbf}{\LWR@HTMLtextbf}
```

`\textrm` $\langle text \rangle$

```
6990 \DeclareRobustCommand{\LWR@HTMLtextrm}[1]{\LWR@htmlspanclass{textrm}{#1}}
6991 \DeclareRobustCommand{\LWR@nulltextrm}[1]{#1}
6992 \LetLtxMacro{\textrm}{\LWR@HTMLtextrm}
```

`\textsf` $\langle text \rangle$

```
6993 \DeclareRobustCommand{\LWR@HTMLtextsf}[1]{\LWR@htmlspanclass{textsf}{#1}}
6994 \DeclareRobustCommand{\LWR@nulltextsf}[1]{#1}
6995 \LetLtxMacro{\textsf}{\LWR@HTMLtextsf}
```

`\texttt` $\langle text \rangle$

```
6996 \DeclareRobustCommand{\LWR@HTMLtexttt}[1]{\LWR@htmlspan{kbd}{#1}}
6997 \DeclareRobustCommand{\LWR@nulltexttt}[1]{#1}
6998 \LetLtxMacro{\texttt}{\LWR@HTMLtexttt}
```

`\textup` $\langle text \rangle$

```
6999 \DeclareRobustCommand{\LWR@HTMLtextup}[1]{\LWR@htmlspanclass{textup}{#1}}
7000 \DeclareRobustCommand{\LWR@nulltextup}[1]{#1}
7001 \LetLtxMacro{\textup}{\LWR@HTMLtextup}
```

`\textit` $\langle text \rangle$

```
7002 \DeclareRobustCommand{\LWR@HTMLtextit}[1]{\LWR@htmlspan{i}{#1}}
7003 \DeclareRobustCommand{\LWR@nulltextit}[1]{#1}
7004 \LetLtxMacro{\textit}{\LWR@HTMLtextit}
```

`\textsc` $\langle text \rangle$

```
7005 \DeclareRobustCommand{\LWR@HTMLtextsc}[1]{\LWR@htmlspanclass{textsc}{#1}}
7006 \DeclareRobustCommand{\LWR@nulltextsc}[1]{#1}
7007 \LetLtxMacro{\textsc}{\LWR@HTMLtextsc}
```

`\textsl` $\langle text \rangle$

```

7008 \DeclareRobustCommand{\LWR@HTMLtextsl}[1]{\LWR@htmlspanclass{testsl}{#1}}
7009 \DeclareRobustCommand{\LWR@nulltextsl}[1]{#1}
7010 \LetLtxMacro{\textsl}{\LWR@HTMLtextsl}

```

`\textnormal` $\langle text \rangle$

```

7011 \DeclareRobustCommand{\LWR@HTMLtextnormal}[1]{\textmd{\textrm{\textup{#1}}}}
7012 \DeclareRobustCommand{\LWR@nulltextnormal}[1]{#1}
7013 \LetLtxMacro{\textnormal}{\LWR@HTMLtextnormal}

```

```

7014 \DeclareRobustCommand{\LWR@nullrmfamily}{}
7015 \DeclareRobustCommand{\LWR@nullsffamily}{}
7016 \DeclareRobustCommand{\LWR@nullttfamily}{}
7017 \DeclareRobustCommand{\LWR@nullbfseries}{}
7018 \DeclareRobustCommand{\LWR@nullmdseries}{}
7019 \DeclareRobustCommand{\LWR@nullupshape}{}
7020 \DeclareRobustCommand{\LWR@nullslshape}{}
7021 \DeclareRobustCommand{\LWR@nullscshape}{}
7022 \DeclareRobustCommand{\LWR@nullitshape}{}
7023 \DeclareRobustCommand{\LWR@nullem}[1]{}
7024 \DeclareRobustCommand{\LWR@nullnormalfont}{}

```

`\LWR@nullfonts` Removes formatting during filename operations.

```

7025 \newcommand*{\LWR@nullfonts}{%
7026 \LetLtxMacro{\emph}{\LWR@nullemph}%
7027 \LetLtxMacro{\textmd}{\LWR@nulltextmd}%
7028 \LetLtxMacro{\textbf}{\LWR@nulltextbf}%
7029 \LetLtxMacro{\textrm}{\LWR@nulltextrm}%
7030 \LetLtxMacro{\textsf}{\LWR@nulltextsf}%
7031 \LetLtxMacro{\texttt}{\LWR@nulltexttt}%
7032 \LetLtxMacro{\textup}{\LWR@nulltextup}%
7033 \LetLtxMacro{\textit}{\LWR@nulltextit}%
7034 \LetLtxMacro{\textsc}{\LWR@nulltextsc}%
7035 \LetLtxMacro{\textsl}{\LWR@nulltextsl}%
7036 \LetLtxMacro{\textnormal}{\LWR@nulltextnormal}%
7037 \LetLtxMacro{\rmfamily}{\LWR@nullrmfamily}%
7038 \LetLtxMacro{\sffamily}{\LWR@nullsffamily}%
7039 \LetLtxMacro{\ttfamily}{\LWR@nullttfamily}%
7040 \LetLtxMacro{\bfseries}{\LWR@nullbfseries}%
7041 \LetLtxMacro{\mdseries}{\LWR@nullmdseries}%
7042 \LetLtxMacro{\upshape}{\LWR@nullupshape}%
7043 \LetLtxMacro{\slshape}{\LWR@nullslshape}%
7044 \LetLtxMacro{\scshape}{\LWR@nullscshape}%
7045 \LetLtxMacro{\itshape}{\LWR@nullitshape}%
7046 \LetLtxMacro{\em}{\LWR@nullem}%
7047 \LetLtxMacro{\normalfont}{\LWR@nullnormalfont}%
7048 \renewcommand*{\HTMLunicode}[1]{}%

```

```
7049 \renewcommand*{\HTMLentity}[1]{}%
```

Ampersand becomes “and”, which is a short word and is then removed from the filename.

```
7050 \renewcommand*{\&}{and}%
7051 \renewcommand{\textsuperscript}[1]{##1}%
7052 \renewcommand{\textsubscript}[1]{##1}%
7053 \LetLtxMacro\underline\LWR@origunderline%
7054 \RenewDocumentCommand{\LWR@htmlspanclass}{o m +m}{##3}%
7055 \DeclareExpandableDocumentCommand{\InlineClass}{+o +m +m}{##3}%
7056 }
```

`\mdseries`

```
7057 \renewcommand*{\mdseries}{}%
```

`\bfseries`

```
7058 \renewcommand*{\bfseries}{}%
```

`\rmfamily`

```
7059 \renewcommand*{\rmfamily}{}%
```

`\sffamily`

```
7060 \renewcommand*{\sffamily}{}%
```

`\ttfamily`

```
7061 \renewcommand*{\ttfamily}{}%
```

`\upshape`

```
7062 \renewcommand*{\upshape}{}%
```

`\itshape`

```
7063 \renewcommand*{\itshape}{}%
```

`\scshape`

```
7064 \renewcommand*{\scshape}{}%
```

`\normalfont`

```
7065 \renewcommand*\normalfont{-}
```

`\sp` $\langle text \rangle$

For siunitx. Must work in math mode.

```
7066 \renewcommand{\sp}[1]{\text{<sup>#1</sup>-}}
```

`\sb` $\langle text \rangle$

For siunitx. Must work in math mode.

```
7067 \renewcommand{\sb}[1]{\text{<sub>#1</sub>-}}
```

`\textsuperscript` $\langle text \rangle$

```
7068 \renewcommand{\textsuperscript}[1]{\LWR@htmlspan{sup}{#1}}
```

`\textsubscript` $\langle text \rangle$

```
7069 \renewcommand{\textsubscript}[1]{\LWR@htmlspan{sub}{#1}}
```

`\up` $\langle text \rangle$ Prints superscript.

This is `\let` at the beginning of the document in case some other package has changed the definition.

```
7070 \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.

```
7071 \AtBeginDocument{\let\fup\textsuperscript}
```

`\underline` $\langle text \rangle$

```
7072 \renewcommand{\underline}[1]{\InlineClass{underline}{#1}}
```

`\hfill`

```
7073 \renewcommand*{\hfill}{\quad}
```

`\hrulefill`

```
7074 \renewcommand*{\hrulefill}{\rule{1in}{1pt}}
```

`\dotfill`

```
7075 \renewcommand*{\dotfill}{\dots}
```

```
7076 \end{warpHTML}
```

73 Skips, spaces, font sizes

for HTML output: 7077 `\begin{warpHTML}`

`\`, must be redefined after `\RequirePackage{printlen}`

```
7078 \let\LWR@origcomma\,
7079 \let\LWR@origtilde~
7080 \let\LWR@origenskip\enskip
7081 \let\LWR@origquad\quad
7082 \let\LWR@origqqquad\qqquad
7083 \let\LWR@orighspace\hspace
7084 \let\LWR@origvspace\vspace
7085 \let\LWR@origrule\rule
7086 \let\LWR@origmedskip\medskip
7087 \let\LWR@origtextellipsis\textellipsis
```

Direct-formatting space commands become HTML entities:

```
7088 \renewcommand*{\,}{\HTMLUnicode{202f}} % HTML thin non-breakable space
```

```
7089
```

```
7090 \renewcommand*{~}{\HTMLentity{nbsp}}
```

```
7091
```

```
7092 \renewcommand*{\textellipsis}{\HTMLUnicode{2026}}
```

Direct-formatting font sizes are ignored:

```
7093 \let\LWR@orignormalsize\normalsize
```

```
7094 \let\LWR@origsmall\small
```

```

7095 \let\LWR@origfootnotesize\footnotesize
7096 \let\LWR@origscriptsize\scriptsize
7097 \let\LWR@origtiny\tiny
7098 \let\LWR@origlarge\large
7099 \let\LWR@origLarge\Large
7100 \let\LWR@origLARGE\LARGE
7101 \let\LWR@orighuge\huge
7102 \let\LWR@origHuge\Huge
7103 \renewcommand*\normalsize{}
7104 \renewcommand*\small{}
7105 \renewcommand*\footnotesize{}
7106 \renewcommand*\scriptsize{}
7107 \renewcommand*\tiny{}
7108 \renewcommand*\large{}
7109 \renewcommand*\Large{}
7110 \renewcommand*\LARGE{}
7111 \renewcommand*\huge{}
7112 \renewcommand*\Huge{}
7113
7114 \renewcommand*\onecolumn{}
7115
7116 \renewcommand{\twocolumn}[1][ ]{
7117
7118 #1
7119
7120 }

```

`\newline` Uses the HTML `
` element.

```

7121 \newcommand*\LWR@newlinebr{\unskip\LWR@htmltag{br /}\LWR@orignewline}%
7122 \let\newline\LWR@newlinebr

```

`\\` Redefined to `\LWR@endofline` or `\LWR@tabularendofline`.

`\LWR@endofline` * [*len*]

`\\` is assigned to `\LWR@endofline` at `\LWR@LwarpStart`.

Inside tabular, `\\` is temporarily changed to `\LWR@tabularendofline`.

```

7123 \LetLtxMacro\LWR@origendofline\\
7124 \NewDocumentCommand{\LWR@endofline}{s o}
7125 {%
7126 \newline%
7127 }

```

`\LWR@minipagestartpars` Minipages are often placed side-by-side inside figures, with a bit of horizontal space to separate them. Since HTML does not allow a `<div>` to be inside a `p`, paragraphs

must be turned off during the generation of the minipage, then turned on after the minipage is complete. When this occurs between side-by-side minipages, `lwarp` correctly suppresses the paragraph tags between the minipages, unless some other text is between the minipages. Such text forms its own paragraph, resulting in text after a minipage to be on its own line. Since people often place small horizontal space between minipages, it is desirable to maintain this space if possible. `lwarp` tries to do this by remembering that a minipage has been seen, in which case paragraph tags are suppressed around `\hspace`, `\enskip`, `\quad`, and `\qquad` until the end of the paragraph, when the closing `p` tag is created.

When a minipage is seen, the boolean `LWR@minipagethispar` is set, telling the following horizontal whitespace commands to try to suppress their surrounding paragraph tags. `LWR@minipagethispar` is cleared at the next end of paragraph, when the HTML paragraph closing tag is generated.

Placed just before `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
7128 \newcommand*{\LWR@minipagestartpars}{%
7129 \ifbool{LWR@minipagethispar}{\LWR@startpars}{}%
7130 }
```

`\LWR@minipagestoppars` Placed just after `\hspace`, `\quad`, or `\qquad`'s HTML output.

```
7131 \newcommand*{\LWR@minipagestoppars}{%
7132 \ifbool{LWR@minipagethispar}{\LWR@stoppars}{}%
7133 }
```

`\quad` Handles special minipage & horizontal space interactions.

```
7134 \renewcommand*{\quad}{%
7135 \LWR@minipagestoppars%
7136 \HTMLUnicode{2001}%
7137 \LWR@minipagestartpars%
7138 }
```

`\qquad` Handles special minipage & horizontal space interactions.

```
7139 \renewcommand*{\qquad}{\quad\quad}
```

`\enskip` Handles special minipage & horizontal space interactions.

```
7140 \renewcommand*{\enskip}{%
7141 \LWR@minipagestoppars%
7142 \HTMLUnicode{2000}%
7143 \LWR@minipagestartpars%
7144 }
```

Len `\WR@tempwidth` Used to compute span width, height, raise for `\hspace` and `\rule`:
 Len `\WR@tempheight` 7145 `\newlength{\LWR@tempwidth}`
 Len `\WR@tempraize` 7146 `\newlength{\LWR@tempheight}`
 7147 `\newlength{\LWR@tempraize}`

`\LWR@hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

Prints a span of a given width. Ignores the optional star.

`\hspace{\fill}` is converted to `\hspace{2em}`, equal to `\quad`.

```
7148 \NewDocumentCommand{\LWR@hspace}{s m}{%
7149 \setlength{\LWR@tempwidth}{#2}%
```

If `\fill`, change to `\quad`:

```
7150 \ifnum\gluestretchorder\LWR@tempwidth>0%
7151 \setlength{\LWR@tempwidth}{2em}%
7152 \fi%
```

Only if the width is not zero:

```
7153 \ifthenelse{\dimtest{\LWR@tempwidth}={}\Opt}{-}{-}{%}
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7154 \LWR@minipagestoppars%
```

Support the HTML thin wrappable space:

```
7155 \ifthenelse{\dimtest{\LWR@tempwidth}={}\{.16667em}}
7156 {%
7157 \HTMLUnicode{2009}% thin breakable space
7158 }%
```

Print the span with the converted width. Not rounded.

```
7159 {%
7160 \uselengthunit{PT}%
7161 \LWR@htmltagc{%
7162 span style="width:\printlength{\LWR@tempwidth}; display:inline-block;"%
7163 }%
7164 \LWR@htmltagc{/span}%
7165 }%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7166 \LWR@minipagestartpars%
7167 }% width not 0
7168 }
```

`\LWR@nohspace` * $\langle length \rangle$

Used to disable `\hspace` while creating description `\items`.

```
7169 \NewDocumentCommand{\LWR@nohspace}{s m}{}

```

`\hspace` * $\langle length \rangle$

Handles special minipage & horizontal space interactions.

```
7170 \LetLtxMacro{\hspace}{\LWR@hspace}

```

`\LWR@vspace` * $\langle length \rangle$ Nullified `vspace`.

```
7171 \NewDocumentCommand{\LWR@vspace}{s m}{}

```

`\vspace` * $\langle length \rangle$ Nullified.

```
7172 \let\vspace\LWR@vspace

```

`\linebreak` [$\langle num \rangle$] Inserts an HTML `br` tag.

```
7173 \renewcommand*{\linebreak}[1] [] {\newline}

```

`\nolinebreak` [$\langle num \rangle$]

```
7174 \renewcommand*{\nolinebreak}[1] [] {}

```

`\pagebreak` [$\langle num \rangle$] Starts a new paragraph.

```
7175 \renewcommand*{\pagebreak}[1] [] {
7176
7177 }
```

`\nopagebreak` [$\langle num \rangle$]

```
7178 \renewcommand*{\nopagebreak}[1] [] {}

```

`\enlargethispage` * $\langle len \rangle$

```
7179 \RenewDocumentCommand{\enlargethispage}{s m}{}

```

`\LWR@currenttextcolor` The color to use for text and `\rule`, defaulting to black:

```
7180 \newcommand*{\LWR@currenttextcolor}{black}

```

`\LWR@rule` [$\langle raise \rangle$] $\langle width \rangle$ $\langle height \rangle$

Handles special minipage & horizontal space interactions.

Creates a span of a given width and height. Ignores the optional star.

`\fill` is zero-width, so `\hspace{\fill}` is ignored.

```
7181 \NewDocumentCommand{\LWR@rule}{o m m}{%

```

The width is copied into a temporary \TeX length, from which comparisons and conversions may be made:

```
7182 \setlength{\LWR@tempwidth}{#2}%

```

If it's zero-width then skip the entire rule:

```
7183 \ifthenelse{\lengthtest{\LWR@tempwidth=0pt}}

```

```
7184 {}% zero- width

```

```
7185 {% non-zero width

```

If it's non-zero width, set a minimal thickness so that it more reliably shows in the browser:

```
7186 \ifthenelse{%
7187 \lengthtest{\LWR@tempwidth>0pt}\AND%
7188 \lengthtest{\LWR@tempwidth<1pt}}%
7189 }%
7190 {\setlength{\LWR@tempwidth}{1pt}}{}%

```

Likewise with height:

```
7191 \setlength{\LWR@tempheight}{#3}%
7192 \ifthenelse{%
7193 \lengthtest{\LWR@tempheight>0pt}\AND%
7194 \lengthtest{\LWR@tempheight<1pt}}%
7195 }%
7196 {\setlength{\LWR@tempheight}{1pt}}{}%

```

If had a minipage this paragraph, try to inline the rule without generating paragraph tags:

```
7197 \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.

```
7198 \uselengthunit{PT}%
7199 \LWR@htmltagc{%
7200 span
7201 style="%
```

The background color is used to draw the filled rule. The color may be changed by `\textcolor`.

```
7202 background:\LWR@currenttextcolor; %
```

The width and height are printed, converted to PT:

```
7203 width:\printlength{\LWR@tempwidth}; %
7204 height:\printlength{\LWR@tempheight}; %
```

The raise height is converted to a css transform. The *2 raise multiplier is to approximately match HTML output's X height. Conversion to a \LaTeX length allows a typical \LaTeX expression to be used as an argument for the raise, whereas printing the raise argument directly to HTML output without conversion to a \LaTeX length limits the allowable syntax. To do: A superior method would compute a ratio of \LaTeX ex height, then print that to HTML with an ex unit.

```
7205 \IfValueT{#1}%
7206 {%
7207 \setlength{\LWR@tempraise}{Opt-#1}%
7208 \setlength{\LWR@tempraise}{\LWR@tempraise*2}%
7209 \LWR@orignewline%
7210 -ms-transform: translate(Opt,\printlength{\LWR@tempraise}); %
7211 \LWR@orignewline%
7212 -webkit-transform: translate(Opt,\printlength{\LWR@tempraise}); %
7213 \LWR@orignewline%
7214 transform: translate(Opt,\printlength{\LWR@tempraise}); %
7215 \LWR@orignewline%
7216 }%
```

Display inline-block to place the span inline with the text:

```
7217 display:inline-block; "%
7218 }%
7219 \LWR@htmltagc{/span}%
```

If had a minipage this paragraph, try to inline the white space without generating paragraph tags:

```
7220 \LWR@minipagestartpars%
7221 }% non-zero width
7222 }
```

```
\rule [⟨raise⟩] {⟨width⟩} {⟨height⟩}
```

Handles special minipage & horizontal space interactions.

```
7223 \renewcommand{\rule}{\LWR@rule}
```

```
7224 \end{warpHTML}
```

74 \phantomsection

for HTML output: 7225 \begin{warpHTML}

\phantomsection Emulate the hyperref \phantomsection command, often used to insert the bibliography into table of contents:

```
7226 \newcommand*{\phantomsection}{%
7227 \section*{}%
7228 }
```

```
7229 \end{warpHTML}
```

75 \LaTeX and other logos

Logos for HTML and print modes:

Some of these logos may be redefined in a later package, so after loading other packages, and at the beginning of the document, their definitions are finally \let in \LWR@LwarpStart.

For css conversions, see:

<http://edward.oconnor.cx/2007/08/tex-poshlet>

<http://nitens.org/taraborelli/texlogo>

75.1 HTML logos

for HTML output: 7230 `\begin{warpHTML}`

`\TeX` \TeX

`latexlogo` is a CSS class used to properly typeset the E and A in $\mathbb{E}\TeX$ and friends.

`latexlogofont` is a CSS class used to select the font for the rest of the logo in $\mathbb{E}\TeX$, `Lua \TeX` , `Con \TeX t`, etc.

```
7231 \let\LWR@origTeX\TeX
7232
7233 \newcommand*{\LWR@TeX}
7234 {\InlineClass{latexlogofont}%
7235 {\InlineClass{latexlogo}{T\textsubscript{e}X}}}
```

`\LaTeX` $\mathbb{E}\TeX$, $\mathbb{E}\TeX 2_{\mathcal{E}}$

`\LaTeXe`

```
7236 \let\LWR@origLaTeX\LaTeX
7237
7238 \newcommand*{\LWR@LaTeX}
7239 {\InlineClass{latexlogofont}%
7240 {\InlineClass{latexlogo}%
7241 {L\textsuperscript{a}T\textsubscript{e}X}}}}
7242
7243 \let\LWR@origLaTeXe\LaTeXe
7244
7245 \renewcommand*{\LaTeXe}
7246 {\LaTeX\InlineClass{latexlogofont}%
7247 {\,\,2\textsubscript{\textit{\HTMLUnicode{3B5}}}}}
```

`\LuaTeX` `Lua \TeX` , `Lua $\mathbb{E}\TeX$`

`\LuaLaTeX`

```
7248 \newcommand*{\LWR@LuaTeX}{\InlineClass{latexlogofont}{Lua}\TeX}
7249 \newcommand*{\LWR@LuaLaTeX}{\InlineClass{latexlogofont}{Lua}\LaTeX}
```

`\XeTeX` `X $\mathbb{E}\TeX$` , `X $\mathbb{E}\mathbb{E}\TeX$`

`\XeLaTeX`

`xetexlogo` is a CSS class which aligns the backwards E in `X $\mathbb{E}\TeX$` and spaces `\TeX` appropriately.

`xelatexlogo` is a CSS class which aligns the backwards E in `X $\mathbb{E}\mathbb{E}\TeX$` and spaces `\mathbb{E}\TeX` appropriately.

```
7250 \newcommand*{\Xe}
```

```

7251 {X\textsubscript{\HTMLunicode{18e}}}
7252 \newcommand*\LWR@XeTeX{\InlineClass{xetexlogo}{\Xe}\TeX}
7253 \newcommand*\LWR@XeLaTeX{\InlineClass{xelatexlogo}{\Xe}\LaTeX}

```

`\ConTeXt` `ConTeXt`

```

7254 \newcommand*\LWR@ConTeXt
7255 {\InlineClass{latexlogofont}{Con}\TeX}%
7256 \InlineClass{latexlogofont}{t}}

```

`\BibTeX` `BIBTEX`, *MakeIndex*
`\MakeIndex`

```

7257 \providecommand*\BibTeX
7258 {\InlineClass{latexlogofont}{B\textsc{ib}}\TeX}
7259
7260 \newcommand*\MakeIndex
7261 {\InlineClass{latexlogofont}{\textit{MakeIndex}}}

```

`\AmS` *AMS*

`amslogo` is a CSS class used for the *AMSlogo*.

```

7262 \AtBeginDocument{\DeclareDocumentCommand{\AmS}{}
7263 {\InlineClass{amslogo}{\textit{A\textsubscript{M}S}}}}

```

`\MiKTeX` `MiKTEX`

```

7264 \newcommand*\MiKTeX{\InlineClass{latexlogofont}{MiK}\TeX}

```

`\LyX` `LyX`

`lyxlogo` is a CSS class used for the *LyXlogo*.

```

7265 \newcommand*\LyX{\InlineClass{lyxlogo}{LyX}}

```

```

7266 \end{warpHTML}

```

75.2 Print logos

```

for PRINT output: 7267 \begin{warpprint}
7268 \newcommand*\XeTeXrevE
7269 {\hspace{-.1667em}\raisebox{-.5ex}{\reflectbox{E}}\hspace{-.125em}}
7270 \providecommand*\XeTeX{\mbox{X\XeTeXrevE\TeX}}
7271 \providecommand*\XeLaTeX{\mbox{X\XeTeXrevE\LaTeX}}

```

```
7272 \providecommand*\AmS}{%
7273 \leavevmode\hbox{$\mathcal A\kern-.2em\lower.376ex%
7274 \hbox{$\mathcal M$}\kern-.2em\mathcal S$}}
7275 \newcommand*\LyX}{\textsf{LyX}}
7276 \providecommand*\LuaTeX}{\mbox{Lua\TeX}}
7277 \providecommand*\LuaLaTeX}{\mbox{Lua\LaTeX}}
7278 \providecommand*\BibTeX}{\mbox{B\textsc{ib}\TeX}}
7279 \providecommand*\MakeIndex}{\mbox{\textit{MakeIndex}}}
7280 \providecommand*\ConTeXt}{\mbox{Con\TeX{t}}}
7281 \providecommand*\MiKTeX}{\mbox{MiK\TeX}}
7282 \end{warpprint}
```

76 \AtBeginDocument, \AtEndDocument

for HTML output: 7283 \begin{warpHTML}

\LWR@LwarpStart Automatically sets up the HTML-related actions for the start and end of the document.

\LWR@LwarpEnd

```
7284 \AfterEndPreamble{\LWR@LwarpStart}
```

```
7285 \AtEndDocument{\LWR@LwarpEnd}
```

```
7286 \end{warpHTML}
```

77 Trademarks

- \TeX is a trademark of American Mathematical Society.
- Adobe® and Adobe Framemaker® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Mac OS® is a trademark of Apple Inc.
- MadCap Flare™ is the property of MadCap Software, Inc.
- MathJax is copyright 2009 and later. The MathJax Consortium is a joint venture of the American Mathematical Society (AMS) and the Society for Industrial and Applied Mathematics (SIAM) to advance mathematical and scientific content on the web.
- Microsoft®, Encarta, MSN, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- UNIX® is a registered trademark of The Open Group.

File 2 **lwarp-a4.sty**

§ 78 Package **a4**

Pkg a4 a4 is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4}
2 \newcommand*{\WideMargins}{}

File 3 **lwarp-a4wide.sty**

§ 79 Package **a4wide**

Pkg a4wide a4wide is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a4wide}

File 4 **lwarp-a5comb.sty**

§ 80 Package **a5comb**

Pkg a5comb a5comb is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{a5comb}

File 5 **lwarp-abstract.sty**

§ 81 Package **abstract**

(Based on original code by PETER WILSON.)

Pkg abstract abstract is supported and patched by lwarp.

 **missing TOC** If using the number option with file splits, be sure to place the table of contents before the abstract. The number option causes a section break which may cause a

file split, which would put a table of contents out of the home page if it is after the abstract.

for HTML output: Accept all options for lwarp-abstract:

```

1 \LWR@ProvidesPackagePass{abstract}

2 \AtBeginDocument{
3 \BeforeBeginEnvironment{abstract}{
4 \LWR@forcenewpage
5 \BlockClass{abstract}
6 }
7 \AfterEndEnvironment{abstract}{\endBlockClass}
8 }
9
10 \renewcommand{\bsrunintitle}{%
11 \hspace*{\abstitlestitle}%
12 {\abstractnamefont%
13 \InlineClass{abstractrunintitle}{\abstractname}%
14 \@bslabeldelim}%
15 }
16
17 \if@titlepage
18 \renewenvironment{abstract}{%
19 % \titlepage
20 \null\vfil
21 \@beginparpenalty\@lowpenalty
22 \if@bsrunin
23 \else
24 \if@bsstyle
25 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
26 \else
27 \ifnumber@bs
28 \num@bs
29 \else
30 \begin{\absnamepos}%
31 \abstractnamefont \BlockClassSingle{abstracttitle}{\abstractname}
32 \@endparpenalty\@M
33 \end{\absnamepos}%
34 %% \vspace{\abstitlestitle}%
35 \fi
36 \fi
37 \vspace{\abstitlestitle}%
38 \fi
39 \put@bsintoc%
40 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
41 {\par\end{@bstr@ctlist}\vfil\null%\endtitlepage
42 }
43 \else
44 \renewenvironment{abstract}{%

```

```

45 \if@bsrunin
46 \else
47 \if@bsstyle
48 \abstitlestyle{\BlockClassSingle{abstracttitle}{\abstractname}}
49 \else
50 \ifnumber@bs
51 \num@bs
52 \else
53 \begin{\absnamepos}%
54 \abstractnamefont\BlockClassSingle{abstracttitle}{\abstractname}%
55 \end\absnamepos%
56 %% \vspace{\abstitleskip}%
57 \fi
58 \fi
59 \vspace{\abstitleskip}%
60 \fi
61 \put@bsintoc%
62 \begin{@bstr@ctlist}\if@bsrunin\@bsrunintitle\fi\abstracttextfont}%
63 {\par\end{@bstr@ctlist}}
64 \fi
65

```

File 6 `lwarp-adjmulticol.sty`

§ 82 Package **adjmulticol**

Pkg `adjmulticol` `adjmulticol` is emulated.

Emulation similar to `multicol` 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}`

Used to compute margins:

3 `\newlength{\LWR@templengthone}%`

4 `\newlength{\LWR@templengthtwo}%`

`adjmulticols` * `{\langle numcols \rangle} {\langle left margi \rangle} {\langle right margin \rangle}`

5 `\NewDocumentEnvironment{adjmulticols}{s m m m}`

6 `{%`

Compute the margins, and limit to positive only:

```
7 \setlength{\LWR@templengthone}{#3}%
8 \ifdimcomp{\LWR@templengthone}{<}{0pt}{\setlength{\LWR@templengthone}{0pt}}{}%
9 \setlength{\LWR@templengthtwo}{#4}
10 \ifdimcomp{\LWR@templengthtwo}{<}{0pt}{\setlength{\LWR@templengthtwo}{0pt}}{}%
```

If one column is specified, use a <div> of class singlecolumn, else use multicols:

```
11 \newcommand*\LWR@mccolstype{multicols}%
12 \ifnumcomp{#2}{=}{1}{\renewcommand*\LWR@mccolstype{singlecolumn}}{}%
```

Help avoid page overflow:

```
13 \LWR@forcenewpage%
```

Print margins in points:

```
14 \uselengthunit{PT}%
```

Create the <div> with the given margin and class:

```
15 \BlockClass[%
16 margin-left:\rndprintlength{\LWR@templengthone} ; %
17 margin-right:\rndprintlength{\LWR@templengthtwo}%
18 ]{\LWR@mccolstype}%
19 }
20 {\endBlockClass}
```

File 7 **lwarp-addlines.sty**

§ 83 Package **addlines**

Pkg addlines addlines is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{addlines}

2 \newcommand*\addlines[1][1]{}
3 \let\addline\addlines
4 \newcommand*\removelines[1][1]{}
5 \let\removeline\removelines
```

File 8 `lwarp-afterpage.sty`§ 84 Package **afterpage**Pkg `afterpage` Emulated.**for HTML output:** Discard all options for `lwarp-afterpage`:

```

1 \LWR@ProvidesPackageDrop{afterpage}
2 \newcommand{\afterpage}[1]{#1}

```

File 9 `lwarp-algorithmicx.sty`§ 85 Package **algorithmicx**Pkg `algorithmicx` `algorithmicx` is supported with minor adjustments.**for HTML output:** `1 \LWR@ProvidesPackagePass{algorithmicx}`

Inside the `algorithmic` environment, level indenting is converted to a `` of the required length, and comments are placed inside a `` which is floated right.

 **package conflicts** If using `\newfloat`, `trivfloat`, and/or `algorithmicx` together, see section [232.1](#).

for HTML output: `2 \begin{warpHTML}`

```

3 \AtBeginEnvironment{algorithmic}{%
4 %
5 \let\origALG@doentity\ALG@doentity%
6 %
7 \renewcommand*\ALG@doentity{%
8 \origALG@doentity%
9 \uselengthunit{PT}%
10 \LWR@htmltagc{%
11 span style="width:\rndprintlength{\ALG@thistlm}; display:inline-block;"%
12 }%
13 \LWR@htmltagc{/span}%
14 }%
15 %
16 \let\origComment\Comment%
17 %
18 \renewcommand{\Comment}[1]{\InlineClass{floatright}{\origComment{#1}}}%

```

```
19 }
20 \end{warpHTML}
```

File 10 **lwarp-alltt.sty**

§ 86 Package **alltt**

Pkg alltt alltt is patched for use by lwarp.

for HTML output:

```
1 \LWR@ProvidesPackagePass{alltt}
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching alltt.}
4 \AtBeginEnvironment{alltt}{%
5 \LWR@forcenewpage
6 \LWR@atbeginverbatim{alltt}\unskip\LWR@origvspace*{-\baselineskip}%
7 }
8 \AfterEndEnvironment{alltt}{\unskip\LWR@origvspace*{-\baselineskip}\LWR@afterendverbatim}
9 }
```

File 11 **lwarp-amsthm.sty**

§ 87 Package **amsthm**

(Based on original code by PUBLICATIONS TECHNICAL GROUP — AMERICAN MATHEMATICAL SOCIETY.)

Pkg amsthm amsthm is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: <div> of class amsthmbody<theoremstyle>
Theorem Name: of class amsthmname<theoremstyle>
Theorem Number: of class amsthmnumber<theoremstyle>
Theorem Note: of class amsthmnote<theoremstyle>
Proof: <div> of class amsthmproof
Proof Name: of class amsthmproofname
 where <theoremstyle> is plain, definition, etc.

for HTML output:

```
1 \LWR@ProvidesPackagePass{amsthm}
```

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \renewcommand{\theoremstyle}[1]{%
4   \ifundefined{th@#1}{%
5     \PackageWarning{amsthm}{Unknown theoremstyle '#1'}%
6     \thm@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8   }{%
9     \thm@style{#1}%
10    \renewcommand{\LWR@newtheoremstyle}{#1}% new
11  }%
12 }
```

Patched to remember the style for this theorem type:

```
13 \def\@xnthm#1#2{%
14   \csedef{\LWR@thmstyle#2}{\LWR@newtheoremstyle}% new
15   \let\@tempa\relax
16   \exp\@ifdefinable\csname #2\endcsname{%
17     \global\exp\let\csname end#2\endcsname\@endtheorem
18     \ifx *#1% unnumbered, need to get one more mandatory arg
19       \edef\@tempa##1{%
20         \gdef\exp\@nx\csname#2\endcsname{%
21           \@nx\@thm{\exp\@nx\csname th@the\thm@style\endcsname}%
22           }{##1}}%
23       \else % numbered theorem, need to check for optional arg
24         \def\@tempa{\@oparg{\@ynthm{#2}}{}}%
25       \fi
26       \AtBeginEnvironment{#2}{\edef\LWR@thisthmstyle{\csuse{\LWR@thmstyle#2}}}% new
27   }%
28   \@tempa
29 }
```

Patched to enclose with css:

```
30 \newcommand{\LWR@haveamsthmname}{
31 \renewcommand{\thmname}[1]{\InlineClass{amsthmname\LWR@thisthmstyle}{##1}}
32 }
33
34 \newcommand{\LWR@haveamsthmnumber}{
35 \renewcommand{\thmnumber}[1]{\InlineClass{amsthmnumber\LWR@thisthmstyle}{##1}}
36 }
37
38 \newcommand{\LWR@haveamsthmnote}{
39 \renewcommand{\thmnote}[1]{\InlineClass{amsthmnote\LWR@thisthmstyle}{##1}}
```

```

40 }
41
42 \LWR@haveamsthmname
43 \LWR@haveamsthmnumber
44 \LWR@haveamsthmnote

```

Patches for css:

```

45 \def\@begintheorem#1#2[#3]{%
46 \LWR@forcenewpage% new
47 \BlockClass{amsthmbody\LWR@thisthmstyle}% new
48 \deferred@thm@head{
49   \the\thm@headfont \thm@indent
50   \@ifempty{#1}{\let\thmname@gobble}{\LWR@haveamsthmname}% new
51   \@ifempty{#2}{\let\thmnumber@gobble}{\LWR@haveamsthmnumber}% new
52   \@ifempty{#3}{\let\thmnote@gobble}{\LWR@haveamsthmnote}% new
53   \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
54   \the\thm@headpunct~
55   \thmheadnl % possibly a newline.
56   \hskip\thm@headsep
57 }%
58 \ignorespaces}

```

Patched for css:

```

59 \def\@endtheorem{\endBlockClass\endtrivlist\@endpefalse }

```

Proof QED symbol:

```

60 \AtBeginDocument{
61 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
62 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
63 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
64 }

```

Patched for css:

```

65 \renewenvironment{proof}[1][\proofname]{\par
66 \LWR@forcenewpage% new
67   \BlockClass{amsthmproof}% new
68   \pushQED{\qed}%
69   \normalfont \topsep6\p@\@plus6\p@\relax
70   \trivlist
71   \item[\hskip\labelsep
72     \InlineClass{amsthmproofname}{#1\@addpunct{.}}]\ignorespaces% changes
73 }{%
74   \InlineClass{theoremendmark}{\popQED}\endtrivlist%
75   \endBlockClass% new
76   \@endpefalse

```

 77 }

File 12 **lwarp-anonchap.sty**§ 88 Package **anonchap**

Pkg anonchap anonchap is emulated.

 **tocloft & other packages** If using tocloft with 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\@secntformat%
10 }
```

File 13 **lwarp-anysize.sty**§ 89 Package **anysize**

Pkg anysize anysize is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{anysize}

2 \def\papersize#1#2{}
3 \def\marginsize#1#2#3#4{}
```

File 14 **lwarp-appendix.sty**§ 90 Package **appendix**

Pkg appendix appendix is patched for use by lwarp.

 **incorrect TOC link** During HTML conversion, the option `toc` without the option `page` results in a TOC link to whichever section was before the `appendices` environment. It is recommended to use both `toc` and also `page` at the same time.

for HTML output:

```

1 \LWR@ProvidesPackagePass{appendix}

2 \renewcommand*{\@chap@pppage}{%
3 \part*{\appendixpagename}
4 \if@dotoc@pp
5 \addappheadtotoc
6 \fi
7 }
8
9 \renewcommand*{\@sec@pppage}{%
10 \part*{\appendixpagename}
11 \if@dotoc@pp
12 \addappheadtotoc
13 \fi
14 }
```

File 15 `lwarp-arabicfront.sty`

§ 91 Package **arabicfront**

Pkg `arabicfront` `arabicfront` is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{arabicfront}`

File 16 `lwarp-authblk.sty`

§ 92 Package **authblk**

Pkg `authblk` `authblk` is patched for HTML.

package support `lwarp` supports the native \TeX titling commands, and also supports the packages `authblk` and `titling`. If both are used, `authblk` should be loaded before `titling`.

 **load order**

`\published` and `\subtitle` If using the titling package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 52.7.

(Based on original code by PATRICK W. DALY.)

for HTML output: Require that `authblk` be loaded before titling:

```

1 \@ifpackageloaded{titling}{
2 \PackageError{lwarp-authblk}
3 {Package authblk must be loaded before titling}
4 {Titling appends authblk's author macro, so authblk must be loaded first.}
5 }
6 {}

```

Load authblk:

```
7 \LWR@ProvidesPackagePass{authblk}
```

Patch to add a class for the affiliation:

```

8 \LetLtxMacro\LWRAB@affil\affil
9
10 \renewcommand{\affil}[2] [] {%
11 \LWRAB@affil[#1]{\protect\InlineClass{affiliation}{#2}}
12 }

```

Create an HTML break for an \authorcr:

```
13 \renewcommand*{\authorcr}{\protect\LWR@newlinebr}
```

File 17 **lwarp-balance.sty**

§ 93 Package **balance**

Pkg balance Emulated.

for HTML output: Discard all options for lwarp-balance:

```

1 \LWR@ProvidesPackageDrop{balance}

2 \newcommand*{\balance}{}
3 \newcommand*{\nobalance}{}

```

File 18 **lwarp-bigdelim.sty**

§ 94 Package **bigdelim**

Pkg bigdelim bigdelim is used as-is for print or lateximage, and patched for HTML.

The delimiters are displayed in HTML by printing the delimiter, the text, and a thick border across the side of the \multirow which indicates the actual height of the

delimiter. The delimiter character is given a `` class of `ldelim` or `rdelim`, and the default CSS sets this to `font-size:200%`

⚠ use `\mrowcell` `\ldelim` and `\rdelim` use `\multirow`, so `\mrowcell` must be used in the proper number of empty cells in the same column below `\ldelim` or `\rdelim`, but not in cells which are above or below the delimiter:

```
\begin{tabular}{lll}
<empty> & a & b \\
\ldelim{\}{2}{.25in}[left ] & c & d \\
\mrowcell & e & f \\
<empty> & g & h \\
\end{tabular}
```

```

      a  b
left { c  d
     e  f
      g  h

```

for HTML output:

First, remove the temporary definitions of `\ldelim` and `\rdelim`, which were previously defined for tabular scanning in case `bigdelim` was not loaded:

```
1 \let\ldelim\relax
2 \let\rdelim\relax
```

Next, load the package's new definitions:

```
3 \LWR@ProvidesPackagePass{bigdelim}
```

Remember the print-mode versions:

```
4 \LetLtxMacro\LWR@origldelim\ldelim
5 \LetLtxMacro\LWR@origrdelim\rdelim
```

```
\ldelim  {\langle 1:delimiter \rangle} {\langle 2:#rows \rangle} {\langle 3:width \rangle} [\langle 4:text \rangle]
\rdelim

6 \RenewDocumentCommand{\ldelim}{m m m O{}}{%
7 \renewcommand{\LWR@multirowborder}{right}%
8 \multirow{#2}{#3}{#4 \InlineClass{ldelim}{#1}}%
9 }
10
11 \RenewDocumentCommand{\rdelim}{m m m O{}}{%
12 \renewcommand{\LWR@multirowborder}{left}%
13 \multirow{#2}{#3}{\InlineClass{rdelim}{#1} #4}%
14 }
```

When entering a `lateximage`, restore the print-mode versions:

```

15 \appto{\LWR@restoreorigformatting}{%
16 \LetLtxMacro{\ldelim}{\LWR@origldelim}%
17 \LetLtxMacro{\rdelim}{\LWR@origrdelim}%
18 }

```

File 19 **lwarp-bigstrut.sty**

§ 95 Package **bigstrut**

Pkg bigstrut bigstrut is used as-is for print or lateximage, and patched for HTML.

for HTML output:

```

1 \LWR@ProvidesPackagePass{bigstrut}

2 \LetLtxMacro\LWR@origbigstrut\bigstrut
3
4 \renewcommand\bigstrut[1][x]{}
5
6 \appto{\LWR@restoreorigformatting}{%
7 \LetLtxMacro{\bigstrut}{\LWR@origbigstrut}%
8 }

```

File 20 **lwarp-bookmark.sty**

§ 96 Package **bookmark**

Pkg bookmark bookmark is emulated during HTML output, and the bookmark package is ignored.

for HTML output: Discard all options for lwarp-bookmark:

```

1 \LWR@ProvidesPackageDrop{bookmark}

2 \newcommand*\bookmarksetup[1]{}
3 \newcommand*\bookmarksetupnext[1]{}
4 \newcommand*\bookmark[2][]{}
5 \newcommand*\bookmarkdefinestyle[2]{}
6 \newcommand*\bookmarkget[1]{}
7 \newcommand*\BookmarkAtEnd[1]{}

```

File 21 `lwarp-booktabs.sty`

§ 97 Package **booktabs**

Pkg `booktabs` `booktabs` is emulated during HTML output, and used as-is during print output and inside an HTML `lateximage`.

for HTML output: `1 \LWR@ProvidesPackagePass{booktabs}`

Booktabs emulation is spread among the tabular code. The original definitions are saved here for use in HTML `lateximages`. The HTML versions temporarily overwrite these print versions when `tabular` is started.

```
2 \LetLtxMacro\LWR@origtoprule\toprule
3 \LetLtxMacro\LWR@origmidrule\midrule
4 \LetLtxMacro\LWR@origcmidrule\cmidrule
5 \LetLtxMacro\LWR@origbottomrule\bottomrule
6 \LetLtxMacro\LWR@origaddlinespace\addlinespace
7 \LetLtxMacro\LWR@origmorecmidrules\morecmidrules
8 \LetLtxMacro\LWR@origspecialrule\specialrule
```

File 22 `lwarp-boxedminipage.sty`

§ 98 Package **boxedminipage**

Pkg `boxedminipage` `boxedminipage` is superseded by `boxedminipage2e`.

for HTML output: `1 \LWR@loadnever{boxedminipage}{boxedminipage2e}`

File 23 `lwarp-boxedminipage2e.sty`

§ 99 Package **boxedminipage2e**

Pkg `boxedminipage2e` `boxedminipage2e` is emulated.

for HTML output: Discard all options for `lwarp-boxedminipage2e`:

```
1 \LWR@ProvidesPackageDrop{boxedminipage2e}
```

```

2 \newenvironment{boxedminipage}{%
3 \begin{BlockClass}{framebox}%
4 \minipage%
5 }
6 {
7 \endminipage%
8 \end{BlockClass}
9 }

```

File 24 `lwarp-caption2.sty`

§ 100 Package **caption2**

Pkg `caption2` `caption2` is not used. The user is recommended to use `caption` instead.

for HTML output: `1 \LWR@loadnever{caption2}{caption}`

File 25 `lwarp-ccaption.sty`

§ 101 Package **ccaption**

Pkg `ccaption` `ccaption` is not used. The user is recommended to use `caption` instead.

for HTML output: `1 \LWR@loadnever{ccaption}{caption}`

File 26 `lwarp-change page.sty`

§ 102 Package **change page**

Pkg `change page` `change page` is ignored.

for HTML output: Discard all options for `lwarp-change page`:

```

1 \LWR@ProvidesPackageDrop{change page}

2 \newif\ifoddpge
3 \DeclareRobustCommand{\checkoddpge}{\oddpagetrue}
4 \DeclareRobustCommand{\changetext}[5]{}
5 \DeclareRobustCommand{\change page}[9]{}
6 \newenvironment{adjustwidth}[2]{}{}
7 \newenvironment{adjustwidth*}[2]{}{}

```

File 27 **lwarp-chngpage.sty**

§ 103 Package **chngpage**

Pkg chngpage chngpage is superceded by changepage.

for HTML output: 1 \LWR@loadnever{chngpage}{changepage}

File 28 **lwarp-chappg.sty**

§ 104 Package **chappg**

Pkg chappg chappg is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{chappg}

2 \renewcommand{\pagenumbering}[2] [] {}

3 \providecommand{\chappgsep}{--}

File 29 **lwarp-color.sty**

§ 105 Package **color**

Pkg color Allowed but ignored. xcolor is then required as well.

color is superceded by xcolor, and lwarp requires several of the features of xcolor.

 **missing colors** It should be sufficient for the user's document to load color then load xcolor as well.

for HTML output: 1 \LWR@ProvidesPackagePass{color}

2 \RequirePackage{xcolor}

File 30 **lwarp-crop.sty**

§ 106 Package **crop**

Pkg crop Emulated.

for HTML output: Discard all options for lwarp-crop:

```
1 \LWR@ProvidesPackageDrop{crop}
2 \newcommand*{\crop}[1] [] {}
3 \newcommand*{\cropdef}[6] [] {}
```

File 31 **lwarp-cuted.sty**

§ 107 Package **cuted**

Pkg cuted cuted is emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{cuted}
2 \newenvironment{strip}{}{}
3 \newskip\stripsep
4 \def\oldcolsbreak#1{}
```

File 32 **lwarp-cutwin.sty**

§ 108 Package **cutwin**

Pkg cutwin Emulated.

for HTML output: Discard all options for lwarp-cutwin:

```
1 \LWR@ProvidesPackageDrop{cutwin}
2 \newcommand*{\opencutleft}{}
3 \newcommand*{\opencutright}{}
4 \newcommand*{\opencutcenter}{}
5 \newcommand*{\cutfuzz}{}
6
7 \newenvironment{cutout}[4]
8 {\marginpar{\windowpagestuff}}
9 {}
10
11 \newcommand*{\windowpagestuff}{}
12
13 \newcommand*{\pageinwindow}{%
14 % \begin{minipage}{.3\linewidth}
15 \windowpagestuff
16 % \end{minipage}}
```

```

17 }
18
19 \newenvironment{shapedcutout}[3]
20 {\marginpar{\picinwindow}}
21 {}
22
23 \newcommand*\putstuffinpic{}
24
25 \newcommand*\picinwindow{%
26 \begin{picture}(0,0)
27 \putstuffinpic
28 \end{picture}}

```

File 33 **lwarp-dblfnote.sty**

§ 109 Package **dblfnote**

Pkg dblfnote dblfnote is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{dblfnote}

```

2 \newcounter{DFNsloppiness}
3 \newdimen\DFNcolumnsep
4 \newdimen\DFNcolumnwidth
5 \def\DFNallowcbreak{}
6 \def\DFNinhibitcbreak{}
7 \def\DFNtrysingle{}
8 \def\DFNalwaysdouble{}
9 \def\DFNruleboth{}
10 \def\DFNruleleft{}

```

File 34 **lwarp-dcolumn.sty**

§ 110 Package **dcolumn**

Pkg dcolumn dcolumn is emulated during HTML output, and the dcolumn package is ignored.

1 \LWR@ProvidesPackageDrop{dcolumn}

 File 35 **lwarp-draftwatermark.sty**

 § 111 Package **draftwatermark**

Pkg draftwatermark draftwatermark is emulated during HTML output, and the draftwatermark package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{draftwatermark}

2 \newcommand{\SetWatermarkAngle}[1]{}
3 \newcommand{\SetWatermarkColor}[1]{}
4 \newcommand{\SetWatermarkLightness}[1]{}
5 \newcommand{\SetWatermarkFontSize}[1]{}
6 \newcommand{\SetWatermarkScale}[1]{}
7 \newcommand{\SetWatermarkHorCenter}[1]{}
8 \newcommand{\SetWatermarkVertCenter}[1]{}
9 \newcommand{\SetWatermarkText}[1]{}

```

 File 36 **lwarp-ebook.sty**

 § 112 Package **ebook**

Pkg ebook ebook is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{ebook}

2 \setcounter{secnumdepth}{0}
3 \setcounter{tocdepth}{2}
4
5 \providecommand{\pagefill}[1][0.001mm]{\noindent}
6
7 \providecommand{\ebook}{
8 \setcounter{secnumdepth}{0}
9 \setcounter{tocdepth}{2}
10 }

```

File 37 **lwarp-ellipsis.sty**

§ 113 Package **ellipsis**

Pkg `ellipsis` `ellipsis` is emulated during HTML output, and the `ellipsis` package is ignored.

```
1 \LWR@ProvidesPackageDrop{ellipsis}
2
3 \newcommand{\ellipsisgap}{0.1em}
```

File 38 **lwarp-emptypage.sty**

§ 114 Package **emptypage**

Pkg `emptypage` `emptypage` is ignored.

for HTML output: Discard all options for `lwarp-emptypage`:

```
1 \LWR@ProvidesPackageDrop{emptypage}
```

File 39 **lwarp-endnotes.sty**

§ 115 Package **endnotes**

(Based on original code by JOHN LAVAGNINO.)

Pkg `endnotes` Used as-is.

table of contents To place the endnotes in the TOC, use:

```
\usepackage{endnotes}
\appto\noteheading{\addcontentsline{toc}{section}{\notesname}}
\renewcommand*{\notesname}{Endnotes} % optional
```

HTML page To additionally have the endnotes on their own HTML page, if `FileDepth` allows:

```
\ForceHTMLPage
\theendnotes
```

for HTML output: `1 \LWR@ProvidesPackagePass{endnotes}`

```

2 \def\enoteformat{%
3% \rightskip\z@ \leftskip\z@ \parindent=1.8em
4 \leavevmode
5% \llap{
6 \makeenmark
7% }
8 }
9
10 \def\@makeenmark{\hbox{\LWR@htmlspan{sup}{\normalfont\theenmark}}}
11 \def\makeenmark{\@makeenmark}

```

File 40 **lwarp-enumerate.sty**

§ 116 Package **enumerate**

Pkg enumerate enumerate is supported with no changes.

This package is only required because it was used in the past to drop and then emulate the package. It cannot be removed because an older version which dropped the package may still remain, for example in a local vs. distribution directory, but it is now supported directly by lwarp and thus must no longer be dropped.

for HTML output: 1 \LWR@ProvidesPackagePass{enumerate}

File 41 **lwarp-enumitem.sty**

§ 117 Package **enumitem**

Pkg enumitem enumitem is supported with minor adjustments.

for HTML output: 1 \LWR@ProvidesPackagePass{enumitem}

for HTML output: 2 \begin{warpHTML}

```

\newlist {<name>} {<type>} {<maxdepth>}
\renewlist {<name>} {<type>} {<maxdepth>}

```

For enumitem lists, new lists must have the start and end actions assigned to the new environment. Renewed lists already have their actions assigned, and thus need no changes.

```

3 \let\LWR@orignewlist\newlist
4
5 \renewcommand*{\newlist}[3]{%

```

```

6 \LWR@orignewlist{#1}{#2}{#3}%
7 \AtBeginEnvironment{#1}{\csuse{LWR@#2start}}%
8 \AtEndEnvironment{#1}{\csuse{LWR@#2end}}%
9 }

10 \end{warpHTML}

```

File 42 **lwarp-epigraph.sty**

§ 118 Package **epigraph**

Pkg epigraph epigraph is emulated during HTML output, and the epigraph package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{epigraph}

```

2 \newcommand{\qitem}[2]
3 {
4 \begin{BlockClass}{qitem}
5 #1
6 \begin{BlockClass}{epigraphsource}
7 #2
8 \end{BlockClass}
9 \end{BlockClass}
10 }

11 \newcommand{\epigraph}[2]
12 {
13 \begin{BlockClass}{epigraph}
14 \qitem{#1}{#2}
15 \end{BlockClass}
16 }
17
18 \newenvironment*{epigraphs}
19 {\BlockClass{epigraph}}
20 {\endBlockClass}

```

Use CSS to format epigraphs.

The following are null commands for source compatibility:

```

21 \newlength{\epigraphwidth}
22 \setlength{\epigraphwidth}{.5\linewidth}
23 \newenvironment*{flushepinormal}{}{}
24 \newcommand{\textflush}[1]{flushepinormal}
25 \newcommand{\epigraphflush}[1]{flushright}
26 \newcommand{\sourceflush}[1]{flushright}

```

```

27 \newcommand*\epigraphsize{\small}
28 \newlength{\epigraphrule}
29 \newlength{\beforeepigraphskip}
30 \newlength{\afterepigraphskip}
31 \newcommand{\epigraphhead}[2][0]{#2}
32 \newcommand{\dropchapter}[1]{}
33 \newcommand*\undodrop{}
34 \newcommand{\cleartoevenpage}[1][{}]{}
```

File 43 lwarp-eso-pic.sty

§ 119 Package **eso-pic**

Pkg `eso-pic` `eso-pic` is emulated during HTML output, and the `eso-pic` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{eso-pic}`

```

2 \newcommand*\LenToUnit{}
3 \newcommand{\AtPageUpperLeft}[1]{}
4 \newcommand{\AtPageLowerLeft}[1]{}
5 \newcommand{\AtPageCenter}[1]{}
6 \newcommand{\AtStockLowerLeft}[1]{}
7 \newcommand{\AtStockUpperLeft}[1]{}
8 \newcommand{\AtStockCenter}[1]{}
9 \newcommand{\AtTextUpperLeft}[1]{}
10 \newcommand{\AtTextLowerLeft}[1]{}
11 \newcommand{\AtTextCenter}[1]{}
12 \NewDocumentCommand{\AddToShipoutPictureBG}{s +m}{}
13 \renewcommand{\AddToShipoutPicture}{\AddToShipoutPictureBG}
14 \NewDocumentCommand{\AddToShipoutPictureFG}{s +m}{}
15 \newcommand*\ClearShipoutPictureBG{}
16 \newcommand*\ClearShipoutPicture{}
17 \newcommand*\ClearShipoutPictureFG{}
18 \newcommand{\gridSetup}[6][{}]{}
```

File 44 lwarp-everypage.sty

§ 120 Package **everypage**

Pkg `everypage` `everypage` is emulated during HTML output, and the `everypage` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{everypage}`

```

2 \newcommand*\AddEverypageHook}[1]{}

```

```
3 \newcommand*\AddThispageHook}[1] {}
```

File 45 **lwarp-everyshi.sty**

§ 121 Package **everyshi**

Pkg everyshi Emulated.

for HTML output: Discard all options for lwarp-everyshi:

```
1 \LWR@ProvidesPackageDrop{everyshi}
```

```
2 \newcommand*\EveryShipout}[1] {}
```

```
3 \newcommand*\AtNextShipout}[1] {}
```

File 46 **lwarp-extramarks.sty**

§ 122 Package **extramarks**

Pkg extramarks extramarks is not used.

for HTML output: Discard all options for lwarp-extramarks:

```
1 \LWR@ProvidesPackageDrop{extramarks}
```

```
2 \newcommand*\extramarks}[2] {}
```

```
3 \newcommand*\firstleftxmark {}
```

```
4 \newcommand*\lastleftxmark {}
```

```
5 \newcommand*\firstrightxmark {}
```

```
6 \newcommand*\lastrightxmark {}
```

```
7 \newcommand*\firstxmark {}
```

```
8 \newcommand*\lastxmark {}
```

```
9 \newcommand*\topxmark {}
```

```
10 \newcommand*\topleftxmark {}
```

```
11 \newcommand*\firstleftmark {}
```

```
12 \newcommand*\lastrightmark {}
```

File 47 **lwarp-fancybox.sty**

§ 123 Package **fancybox**

(Based on original code by TIMOTHY VAN ZANDT.)

Pkg `fancybox` `fancybox` is supported with some patches.

framed equation example `fancybox`'s documentation has an example `FramedEqn` environment which combines `math`, `\Sbox`, a `minipage`, and an `\fbox`. This combination requires that the entire environment be enclosed inside a `lateximage`, which is done by adding `\lateximage` at the very start of `FramedEqn`'s beginning code, and `\endlateximage` at the very end of the ending code. Unfortunately, the `HTML alt` attribute is not used here.

```
\newenvironmentFramedEqn
{
\lateximage% NEW
\setlength{\fboxsep}{15pt}
...}{...
\[\fbox{\TheSbox}\]
\endlateximage% NEW
}
```

framing alternatives `\fbox` works with `fancybox`. Also see `lwarp`'s `\fboxBlock` macro and `fminipage` environment for alternatives to `\fbox` for framing environments.

framed table example The `fancybox` documentation's example framed table using an `\fbox` containing a `tabular` does not work with `lwarp`, but the `FramedTable` environment does work if `\fbox` is replaced by `\fboxBlock`. This method loses `HTML` formatting. A better method is to enclose the table's contents inside a `fminipage` environment. The caption may be placed either inside or outside the `fminipage`:

```
\begin{table}
\begin{fminipage}{\linewidth}
\begin{tabular}{lr}
...
\end{tabular}
\end{fminipage}
\end{table}
```

framed verbatim `lwarp` does not support the `verbatim` environment inside a `span`, `box`, or `fancybox`'s `\Sbox`, but a `verbatim` may be placed inside a `fminipage`. The `fancybox` documentation's example `FramedVerb` may be defined as:

```
\newenvironment{FramedVerb}[1] % width
{
\VerbatimEnvironment
\fminipage{#1}
\beginVerbatim
}{
\endVerbatim
\endfminipage
}
```

framed `\VerbBox` fancybox's `\VerbBox` may be used inside `\fbox`.

indented alignment `LVerbatim`, `\LVerbatimInput`, and `\LUseVerbatim` indent with horizontal space which may not line up exactly with what `pdftotext` detects. Some lines may be off slightly in their left edge.

for HTML output:

```

1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{fancybox}

3 \renewcommand*{\@shadowbox}[1]{\InlineClass{shadowbox}{#1}}
4
5 \renewcommand*{\@doublebox}[1]{\InlineClass{doublebox}{#1}}
6
7 \renewcommand*{\@ovalbox}[2]{%
8 \ifthenelse{\isequivalentto{#1}{\thinlines}}
9 {\InlineClass{ovalbox}{#2}}
10 {\InlineClass{Ovalbox}{#2}}
11 }

```

Convert minipages, parboxes, and lists into linear text using the `LWR@nestspan` environment:

```

12 \let\LWR@origSbox\Sbox
13
14 \def\Sbox{\LWR@origSbox\LWR@nestspan}
15
16
17 \let\LWR@origendSbox\endSbox
18
19 \def\endSbox{\endLWR@nestspan\LWR@origendSbox}

```

`Beqnarray` is adapted for MathJax or enclosed inside a `lateximage`:

```

20 \RenewEnviron{Beqnarray}
21 {\LWR@eqnarrayfactor}
22
23 \csgpreto{Beqnarray*}{\boolfalse{LWR@numbereqnarray}}

```

`\GenericCaption` is enclosed in an HTML block:

```

24 \renewcommand{\GenericCaption}[1]{%
25 \LWR@htmlblocktag{figcaption}%
26 #1%
27 \LWR@htmlblocktag{/figcaption}%
28 }

```

`Btrivlist` is enclosed in an HTML block:

```

29 \RenewDocumentEnvironment{Btrivlist}{m o}
30 {\begin{BlockClass}{Btrivlist}\tabular{#1}}
31 {\endtabular\end{BlockClass}}

```

Btrivlist is also neutralized when used inside a span:

```

32 \AtBeginEnvironment{LWR@nestspan}{%
33 \RenewDocumentEnvironment{Btrivlist}{m o}{}}%
34 \RenewDocumentCommand{LWR@origitem}{d()}{LWRFB@origitem}%
35 }

```

lwarp's handling of `\item` is patched to accept fancybox's optional arguments:

```

36 \let\LWRFB@origitemizeitem\LWR@itemizeitem
37 \let\LWRFB@origdescitem\LWR@descitem
38 \LetLtxMacro{LWRFB@origitem}{LWR@origitem}
39
40 \RenewDocumentCommand{LWR@itemizeitem}{d()}{LWRFB@origitemizeitem}
41 \RenewDocumentCommand{LWR@descitem}{d()}{LWRFB@origdescitem}

```

The various boxed lists become regular lists:

```

42 \renewenvironment{Bitemize}[1][\begin{itemize}]{\end{itemize}}
43 \renewenvironment{Benumerate}[1][\begin{enumerate}]{\end{enumerate}}
44 \renewenvironment{Bdescription}[1][\begin{description}]{\end{description}}

```

`\boxput` simply prints one then the other argument, side-by-side instead of above and behind:

```

45 \RenewDocumentCommand{\boxput}{s d() m m}{%
46 \IfBooleanTF{#1}{#3\quad#4}{#4\quad#3}%
47 }

```

Neutralized commands:

```

48 \RenewDocumentCommand{\fancyput}{s d() m}{-}
49 \RenewDocumentCommand{\thisfancyput}{s d() m}{-}
50
51 \RenewDocumentCommand{\fancypage}{m m}{-}
52 \RenewDocumentCommand{\thisfancypage}{m m}{-}
53
54 \def\LandScape#1{}
55 \def\endLandScape{}
56 \def\@Landscape#1#2#3{}
57 \def\endLandscape{}

```

Low-level patches for Verbatim, VerbatimInput, UseVerbatim:

```

58 \let\LWRFB@UseVerbatim\UseVerbatim
59 \renewcommand*\UseVerbatim}[1]{%
60 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
61 \LWRFB@UseVerbatim{#1}%
62 \LWR@afterendverbatim%
63 }
64
65 \let\LWRFB@LUseVerbatim\LUseVerbatim
66
67 \renewcommand*\LUseVerbatim}[1]{%
68 \LWR@atbeginverbatim{LVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
69 \noindent%
70 \LWRFB@LUseVerbatim{#1}%
71 \LWR@afterendverbatim%
72 }
73
74 \def\@BUseVerbatim[#1]#2{%
75 \LWR@atbeginverbatim{BVerbatim}\unskip\LWR@origvspace*{-.5\baselineskip}%%
76 \LWRFB@UseVerbatim{#2}%
77 \LWR@afterendverbatim%
78 }

79 \end{warpHTML}

```

File 48 `lwarp-fancyhdr.sty`

§ 124 Package **fancyhdr**

Pkg fancyhdr fancyhdr is nullified.

for HTML output: Discard all options for lwarp-fancyhdr:

```

1 \LWR@ProvidesPackageDrop{fancyhdr}

2 \newcommand*\fancyhead}[2] [] {}
3 \newcommand*\fancyfoot}[2] [] {}
4 \newcommand*\fancyhf}[2] [] {}
5 \newcommand*\fancypagestyle}[2] {}
6 \newcommand*\lhead}[2] [] {}
7 \newcommand*\chead}[2] [] {}
8 \newcommand*\rhead}[2] [] {}
9 \newcommand*\lfoot}[2] [] {}
10 \newcommand*\cfoot}[2] [] {}
11 \newcommand*\rfoot}[2] [] {}

```

```

12 \newcommand*\headrulewidth{}
13 \newcommand*\footrulewidth{}
14 \newcommand*\fancyheadoffset}[2] [] {}
15 \newcommand*\fancyfootoffset}[2] [] {}
16 \newcommand*\fancyhfoffset}[2] [] {}
17 \newcommand*\iffloatpage}[2] {#2}
18 \newcommand*\ifftopfloat}[2] {#2}
19 \newcommand*\iffbotfloat}[2] {#2}

```

File 49 **lwarp-fancyvrb.sty**

§ 125 Package **fancyvrb**

(Based on original code by TIMOTHY VAN ZANDT.)

Pkg fancyvrb fancyvrb is supported with some patches.

for HTML output: 1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackagePass{fancyvrb}

for HTML output: 3 \begin{warpHTML}

Initial default patch for fancyvrb:

```
4 \fvset{frame=none}%
```

For \VerbatimFootnotes:

```

5 \renewcommand{\VerbatimFootnotes}{
6 \PackageError{lwarp}
7 {Verbatim footnotes are not yet supported by lwarp.}
8 {This may be improved some day.}
9 }

```

After the preamble is loaded, after any patches to Verbatim:

```

10 \AfterEndPreamble{
11 \LWR@traceinfo{Patching Verbatim.}

```

Remember the original definition of Verbatim:

```
12 \let\LWRFV@origVerbatim\Verbatim
```

Env Verbatim Patched to place the environment in a fancyvrb div, and the label in a fancyvrblabel1 div. Also corrects the left margin for line numbers. Also uses VerbatimHTMLWidth to

control placement of line numbers on the right. Aligning the right margin requires knowing the width.

```
13 \renewcommand*\Verbatim}{%
14 \LWR@forcenewpage
15 \LWRFV@origVerbatim%
16 }
```

`\LWR@FVstyle` Holds the style of the verbatim.

```
17 \newcommand*\LWR@FVstyle}{}
```

The following patches to Verbatim are executed at the start and end of the environment, depending on the choice of frame. Original code is from the fancyvrb package.

```
18 \newcommand*\LWR@fvstartnone}{%
19 \LWR@traceinfo{fvstartnone}%
20 \hbox to\z@\LWR@atbeginverbatim[\LWR@FVstyle]{verbatim}}%
21 }
22
23 \newcommand*\LWR@fvendnone}{%
24 \LWR@traceinfo{fvendnone}%
25 \hbox to\z@\LWR@afterendverbatim}%
26 }
27
28 \newcommand*\LWR@fvstartsingle}{%
29 \LWR@traceinfo{fvstartsingle}%
30 \LWR@fvstartnone%
31 \FV@BeginListFrame@Single%
32 }
33
34 \newcommand*\LWR@fvendsingle}{%
35 \LWR@traceinfo{fvendsingle}%
36 \FV@endListFrame@Single%
37 \LWR@fvendnone%
38 }
39
40 \newcommand*\LWR@fvstartline}{%
41 \LWR@traceinfo{fvstartline}%
42 \LWR@fvstartnone%
43 \FV@BeginListFrame@Lines%
44 }
45
46 \newcommand*\LWR@fvendline}{%
47 \LWR@traceinfo{fvendline}%
48 \FV@endListFrame@Lines%
49 \LWR@fvendnone%
```

50 }

The following patches select the start/left/right/end behaviors depending on frame. Original code is from the fancyvrb package.

```

51
52 \newcommand*{\LWR@FVfindcurrentcolor}{%
53 \protect\colorlet{LWR@current@color}{.}%
54 \protect\convertcolorspec{named}{LWR@current@color}{HTML}\LWR@tempcolor%
55 }
56
57 \newcommand*{\LWR@FVtextstyle}{%
58 \LWR@FVfindcurrentcolor
59 \ifdefstring{\LWR@tempcolor}{000000}%
60 {}%
61 {color: \#\LWR@tempcolor ; }%
62 }
63
64
65 \newcommand*{\LWR@FVfindbordercolor}{%
66 \FancyVerbRuleColor%
67 \LWR@FVfindcurrentcolor
68 \color{black}
69 }
70
71 % border width of \FV@FrameRule
72 \newcommand*{\LWR@FVborderstyle}[1]{%
73 padding#1: \strip@pt\dimexpr \FV@FrameSep\relax\relax pt ;
74 \LWR@FVfindbordercolor
75 border#1: \strip@pt\dimexpr \FV@FrameRule\relax\relax pt solid \#\LWR@tempcolor ;
76 }
77
78 \def\FV@Frame@none{%
79 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle}%
80 \let\FV@BeginListFrame\LWR@fvstartnone%
81 \let\FV@LeftListFrame\relax%
82 \let\FV@RightListFrame\relax%
83 \let\FV@endListFrame\LWR@fvendnone}
84
85 \FV@Frame@none% default values
86
87 \def\FV@Frame@single{%
88 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{}}%
89 \let\FV@BeginListFrame\LWR@fvstartsingle%
90 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
91 \let\FV@RightListFrame\FV@RightListFrame@Single%
92 \let\FV@endListFrame\LWR@fvendsingle}
93
94 \def\FV@Frame@lines{%

```

```

95 \renewcommand*{\LWR@FVstyle}{%
96   \LWR@FVtextstyle\LWR@FVborderstyle{-top}\LWR@FVborderstyle{-bottom}%
97 }%
98 \let\FV@BeginListFrame\LWR@fvstartline%
99 \let\FV@LeftListFrame\relax%
100 \let\FV@RightListFrame\relax%
101 \let\FV@EndListFrame\LWR@fvendline}
102
103 \def\FV@Frame@topline{%
104 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-top}}%
105 \let\FV@BeginListFrame\LWR@fvstartline%
106 \let\FV@LeftListFrame\relax%
107 \let\FV@RightListFrame\relax%
108 \let\FV@EndListFrame\LWR@fvendnone}
109
110 \def\FV@Frame@bottomline{%
111 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-bottom}}%
112 \let\FV@BeginListFrame\LWR@fvstartnone%
113 \let\FV@LeftListFrame\relax%
114 \let\FV@RightListFrame\relax%
115 \let\FV@EndListFrame\LWR@fvendline}
116
117 \def\FV@Frame@leftline{%
118 \renewcommand*{\LWR@FVstyle}{\LWR@FVtextstyle\LWR@FVborderstyle{-left}}%
119 % To define the \FV@FrameFillLine macro (from \FV@BeginListFrame)
120 \ifx\FancyVerbFillColor\relax%
121 \let\FV@FrameFillLine\relax%
122 \else%
123 \@tempdima\FV@FrameRule\relax%
124 \multiply\@tempdima-\tw@%
125 \edef\FV@FrameFillLine{%
126 {\noexpand\FancyVerbFillColor{\vrule\@width\number\@tempdima sp}%
127 \kern-\number\@tempdima sp}}%
128 \fi%
129 \let\FV@BeginListFrame\LWR@fvstartnone%
130 \let\FV@LeftListFrame\FV@LeftListFrame@Single%
131 \let\FV@RightListFrame\relax%
132 \let\FV@EndListFrame\LWR@fvendnone}

```

Adds the optional label to the top and bottom edges. Original code is from the fancyvrb package.

```

133 \def\FV@SingleFrameLine#1{%
134   \hbox to\z@{%
135 %     \kern\leftmargin
136     \ifnum#1=\z@\relax
137     \let\FV@Label\FV@LabelBegin
138     \else
139     \let\FV@Label\FV@LabelEnd

```

```

140 \fi
141 \ifx\FV@Label\relax
142 % \FancyVerbRuleColor{\vrule \@width\linewidth \@height\FV@FrameRule}%
143 \else
144 \ifnum#1=\z@
145 % \setbox\z@\hbox{\strut\enspace\FV@LabelBegin\enspace\strut}%
146 \ifx\FV@LabelPositionTopLine\relax
147 \else
148 \LWR@FVfindbordercolor
149 \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
150 \LWR@origtextrm{\FV@LabelBegin}% \textrm preserves emdash
151 \LWR@htmltagc{/div}
152 \fi
153 \else
154 % \setbox\z@\hbox{\strut\enspace\FV@LabelEnd\enspace\strut}%
155 \ifx\FV@LabelPositionBottomLine\relax
156 \else
157 \LWR@FVfindbordercolor
158 \LWR@htmltagc{div class="fancyvrblabel" style="color: \#\LWR@tempcolor"}
159 \LWR@origtextrm{\FV@LabelEnd}
160 \LWR@htmltagc{/div}
161 \fi
162 \fi
163
164 \fi
165 \hss
166 }
167 }

```

Processes each line, adding optional line numbers. Original code is from the fancyvrb package.

```

168 \def\FV@ListProcessLine#1{%
169 \hbox to \hsize{%
170 % \kern\leftmargin
171 \hbox to \VerbatimHTMLWidth {%
172 \ifcvoid{FV@LeftListNumber}{}{\kern 2.5em}%
173 \FV@LeftListNumber%
174 % \FV@LeftListFrame
175 \FancyVerbFormatLine{#1}%
176 \hss%
177 % \FV@RightListFrame
178 \FV@RightListNumber%
179 }%
180 \hss% required to avoid underfull hboxes
181 }
182 }

```

Env **BVerbatim**

```

183 \AtBeginEnvironment{BVerbatim}
184 {
185 \LWR@forcenewpage
186 \LWR@atbeginverbatim{bverbatim}
187
188 }
189
190 \AfterEndEnvironment{BVerbatim}
191 {
192 \leavevmode\par\LWR@origvspace{-\baselineskip}
193 \LWR@afterendverbatim
194 }

```

End of the modifications to make at the end of the preamble:

```

195 } % \AfterEndPreamble

196 \end{warpHTML}

```

File 50 **lwarp-figcaps.sty**

§ 126 Package **figcaps**

Pkg **figcaps** Emulated.

for HTML output: Discard all options for lwarp-figcaps:

```

1 \LWR@ProvidesPackageDrop{figcaps}

2 \newcommand*{\figcapson}{}
3 \newcommand*{\figcapsoff}{}
4 \newcommand*{\printfigures}{}
5 \newcommand*{\figmarkon}{}
6 \newcommand*{\figmarkoff}{}
7 \def\figurecapname{Figure Captions}
8 \def\tablepagename{Tables}
9 \def\figurepagename{Figures}

```

File 51 `lwarp-fix2col.sty`

§ 127 Package **fix2col**

Pkg `fix2col` `fix2col` is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{fix2col}`

File 52 `lwarp-float.sty`

§ 128 Package **float and \newfloat**

Pkg `float` `float` is emulated during HTML output, and the `float` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{float}[2016/03/04]`

See section 61.2 for the `\listof` command.

`\newfloat` `{\langle 1: type \rangle}{\langle 2: placement \rangle}{\langle 3: ext \rangle} [\langle 4: within \rangle]`

Emulates the `\newfloat` command from the `float` package.

“placement” is ignored.

```
2 \NewDocumentCommand{\newfloat}{m m m o}{%
3 \IfValueTF{#4}
4 {\DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}}
5 {\DeclareFloatingEnvironment[fileext=#3]{#1}}
```

`newfloat` package automatically creates the `\listof` command for new floats, but `float` does not, so remove `\listof` here in case it is manually created later.

```
6 \cslet{listof#1s}\relax
7 \cslet{listof#1es}\relax
8 }
```

`\floatname` `{\langle type \rangle}{\langle name \rangle}`

Sets the text name of the float, such as “Figure”.

```
9 \NewDocumentCommand{\floatname}{m +m}{%
```

```
10 \SetupFloatingEnvironment{#1}{name=#2}%
11 }
```

`\floatplacement` $\langle type \rangle$ $\langle placement \rangle$

Float placement is ignored.

```
12 \newcommand*\floatplacement}[2]{%
13 \SetupFloatingEnvironment{#1}{placement=#2}%
14 }
```

`\floatstyle` $\langle style \rangle$

Float styles are ignored.

```
15 \newcommand*\floatstyle}[1]{%
16 }
```

`\restylefloat` * $\langle style \rangle$

Float styles are ignored.

```
17 \NewDocumentCommand{\restylefloat}{s m}{%
18 }
```

File 53 `lwarp-floatflt.sty`

§ 129 Package **floatflt**

Pkg floatflt Emulated.

for HTML output: Discard all options for lwarp-floatflt:

```
1 \LWR@ProvidesPackageDrop{floatflt}
```

Borrowed from the lwarp version of keyfloat:

```
2 \NewDocumentEnvironment{KFLTfloatflt@marginfloat}{0{-1.2ex} m}
3 {% start
4 \LWR@maybeinthisfloat%
5 \LWR@forcenewpage
6 \LWR@stoppars%
7 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
8 \LWR@startpars%
9 \captionsetup{type=#2}%
```

```

10 }
11 {
12 \LWR@htmldivclassend{div}
13 }
14
15 \DeclareDocumentEnvironment{floatingfigure}{o m}
16   {\begin{KFLTfloatflt@marginfloat}{figure}}
17   {\end{KFLTfloatflt@marginfloat}}
18
19 \DeclareDocumentEnvironment{floatingtable}{o +m}
20   {\begin{KFLTfloatflt@marginfloat}{table}#2}
21   {\end{KFLTfloatflt@marginfloat}}

```

File 54 `lwarp-floatpag.sty`

§ 130 Package **floatpag**

Pkg floatpag Emulated.

for HTML output: Discard all options for lwarp-floatpag:

```

1 \LWR@ProvidesPackageDrop{floatpag}

2 \newcommand*{\floatpagestyle}[1]{}
3 \newcommand*{\rotfloatpagestyle}[1]{}
4 \newcommand*{\thisfloatpagestyle}[1]{}

```

File 55 `lwarp-floatrow.sty`

§ 131 Package **floatrow**

Pkg floatrow floatrow is emulated during HTML output, and the floatrow package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{floatrow}`

 **subfig package** When combined with the subfig package, while inside a subfloatrow `\ffigbox` and `\ttabbox` must have the caption in the first of the two of the mandatory arguments.

 **`\FBwidth`, `\FBheight`** The emulation of floatrow does not support `\FBwidth` or `\FBheight`. These values are pre-set to `.3\linewidth` and `2in`. Possible solutions include:

- Use fixed lengths. lwarp will scale the HTML lengths appropriately.

- Use `warpprint` and `warpHTML` environments to select appropriate values for each case.
- Inside a `warpHTML` environment, manually change `\FBwidth` or `\FBheight` before the `\ffigbox` or `\ttabbox`. Use `\FBwidth` or `\FBheight` normally afterwards; it will be used as expected in print output, and will use your custom-selected value in HTML output. This custom value will be used repeatedly, until it is manually changed to a new value.

After everything has loaded, remember whether `subcaption` was loaded. If not, it is assumed that `subfig` is used instead:

```

2 \newbool{LWR@subcaptionloaded}
3
4 \AtBeginDocument{
5 \@ifpackageloaded{subcaption}
6 {\booltrue{LWR@subcaptionloaded}}
7 {\boolfalse{LWR@subcaptionloaded}}
8 }

```

`\floatbox` [*1 preamble*] [*2 captype*] [*3 width*] [*4 height*] [*5 vert pos*] [*6 caption*] [*7 object*]

Only parameters for `captype`, `width`, `caption`, and `object` are used.

`LWR@insubfloatrow` is true if inside a `subfloatrow` environment.

There are two actions, depending on the use of `subcaption` or `subfig`.

```

9 \NewDocumentCommand{\floatbox}{o m o o o +m +m}{%
10 \ifbool{LWR@subcaptionloaded}%
11 {% subcaption

```

For `subcaption`:

```

12   \ifbool{LWR@insubfloatrow}%
13   {% subcaption in a subfloatrow

```

`subfigure` and `subtable` environments take `width` as an argument.

```

14     \IfValueTF{#3}%
15     {\@nameuse{sub#2}{#3}}%
16     {\@nameuse{sub#2}{\linewidth}}%
17   }% subcaption in a subfloatrow
18   {% subcaption not in subfloatrow

```

`figure` and `table` environments do not take a `width` argument.

```

19     \@nameuse{#2}%
20   }% subcaption not in subfloatrow
21   #6
22
23   #7

```

End the environments:

```

24 \ifbool{LWR@insubfloatrow}%
25 {\@nameuse{endsub#2}}%
26 {\@nameuse{end#2}}%
27 }% subcaption
28 {% assume subfig

```

For subfig:

```

29 \ifbool{LWR@insubfloatrow}%
30 {% subfig in a subfloatrow

```

`\subfloat` is a macro, not an environment.

Package `subfig`'s `\subfloat` command takes an optional argument which is the caption, but `\floatbox` argument #6 contains commands to create the caption and label, not the caption itself. Thus, `\caption` is temporarily disabled to return its own argument without braces.

```

31 \begingroup
32 \let\caption\@firstofone
33 \subfloat[#6]{#7}
34 \endgroup
35 }% subfig in a subfloatrow
36 {% subfig package, but not a subfig

```

`figure` and `table` are environments:

```

37 \@nameuse{#2}
38 #6
39
40 #7
41 \@nameuse{end#2}
42 }% subfig package, but not a subfig
43 }% assume subfig
44 }

```

Not used:

```

45 \newcommand*{\nocapbeside}{}
46 \newcommand*{\capbeside}{}
47 \newcommand*{\captop}{}
48 \newlength{\FBwidth}
49 \setlength{\FBwidth}{.3\linewidth}
50 \newlength{\FBheight}
51 \setlength{\FBheight}{2in}
52 \newcommand*{\useFCwidth}{}
53 \newcommand{\floatsetup}[2][{}]{
54 \newcommand{\thisfloatsetup}[1]{
55 \newcommand{\clearfloatsetup}[1]{
56 \newcommand*{\killfloatstyle}{}

```

`\newfloatcommand` [*1 command*] [*2 captype*] [*3 preamble*] [*4 default width*]

Preamble and default width are ignored.

```
57 \NewDocumentCommand{\newfloatcommand}{m m o o}{%
58 \@namedef{#1}{
59 \floatbox{#2}
60 }
61 }
```

`\renewfloatcommand` [*1 command*] [*2 captype*] [*3 preamble*] [*4 default width*]

Preamble and default width are ignored.

```
62 \NewDocumentCommand{\renewfloatcommand}{m m o o}{%
63 \@namedef{#1}{%
64 \floatbox{#2}
65 }
66 }
```

`\ffigbox` [*width*] [*height*] [*vposn*] [*caption commands*] [*contents*]

```
67 \newfloatcommand{ffigbox}{figure}[\nocapbeside] []
```

`\ttabbox` [*width*] [*height*] [*vposn*] [*caption commands*] [*contents*]

```
68 \newfloatcommand{ttabbox}{table}[\capttop] [\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{\buildFBBOX}[2]{}
119 \newcommand*\CenterFloatBoxes{}
120 \newcommand*\TopFloatBoxes{}
121 \newcommand*\BottomFloatBoxes{}
122 \newcommand*\PlainFloatBoxes{}
123
124 \newcommand{\capsubrowsettings}{}
125
126 \NewDocumentCommand{\RawFloats}{o o}{}

```

`\RawCaption` $\langle text \rangle$

To be used inside a minipage or parbox.

```
127 \newcommand{\RawCaption}[1]{#1}
```

`\floatfoot` $\langle text \rangle$

Places additional text inside a float, inside a CSS `<div>` of class `floatfoot`.

```

128 \NewDocumentCommand{\floatfoot}{s +m}{%
129 \begin{BlockClass}{floatfoot}
130 #2
131 \end{BlockClass}}
132 }

```

Used to compute `\linewidth`.

```

133 \newbool{LWR@insubfloatrow}
134 \boolfalse{LWR@insubfloatrow}

```

Env `subfloatrow` $[\langle num_floats \rangle]$

```

135 \newenvironment*{subfloatrow}[1][2]
136 {

```

The row of floats is placed into a `<div>` of class `floatrow`:

```

137 \LWR@forcenewpage
138 \BlockClass{floatrow}

```

While inside the `floatrow`, `LWR@insubfloatrow` is set true, which tells `\floatbox` to use `\subfigure` or `\subtable`.

```
139 \beginingroup
```

```

140 \booltrue{LWR@insubfloatrow}
141 }
142 {
143 \endgroup
144 \endBlockClass
145 \boolfalse{LWR@insubfloatrow}
146 }

```

File 56 **lwarp-flushend.sty**

§ 132 Package **flushend**

Pkg flushend Emulated.

for HTML output: Discard all options for lwarp-flushend:

```

1 \LWR@ProvidesPackageDrop{flushend}
2 % \end{ma-crocode}
3 %
4 % \begin{macrocode}
5 \newcommand*{\flushend}{}
6 \newcommand*{\raggedend}{}
7 \newcommand*{\flushcolsend}{}
8 \newcommand*{\raggedcolsend}{}
9 \newcommand*{\atColsBreak}[1]{}
10 \newcommand*{\atColsEnd}[1]{}
11 \newcommand*{\showcolsendrule}{}

```

File 57 **lwarp-fncychap.sty**

§ 133 Package **fncychap**

Pkg fncychap fncychap is ignored.

for HTML output: Discard all options for lwarp-fncychap:

```

1 \LWR@ProvidesPackageDrop{fncychap}

2 \def\mghrulefill#1{}
3 \def\ChNameLowerCase{}
4 \def\ChNameUpperCase{}
5 \def\ChNameAsIs{}
6 \def\ChTitleLowerCase{}
7 \def\ChTitleUpperCase{}
8 \def\ChTitleAsIs{}

```

```

9 \newcommand{\ChRuleWidth}[1]{}
10 \newcommand{\ChNameVar}[1]{}
11 \newcommand{\ChNumVar}[1]{}
12 \newcommand{\ChTitleVar}[1]{}
13 \newcommand{\TheAlphaChapter}{}
14 \newcommand{\DOCH}{}
15 \newcommand{\DOTI}[1]{}
16 \newcommand{\DOTIS}[1]{}
17 \newlength{\mylen}
18 \newlength{\myhi}
19 \newlength{\px}
20 \newlength{\py}
21 \newlength{\pyy}
22 \newlength{\pxx}
23 \newlength{\RW}
24 \newcommand{\FmN}[1]{#1}
25 \newcommand{\FmTi}[1]{#1}

```

File 58 **lwarp-fnpos.sty**

§ 134 Package **fnpos**

Pkg `fnpos` `fnpos` is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fnpos}

2 \newcommand*{\makeFNbottom}{}
3 \newcommand*{\makeFNmid}{}
4 \newcommand*{\makeFNbelow}{}
5 \newcommand*{\makeFNabove}{}

```

File 59 **lwarp-fontenc.sty**

§ 135 Package **fontenc**

Pkg `fontenc` Error if `fontenc` is loaded after `lwarp`.

Discard all options for `lwarp-fontenc`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fontenc}

2 \LWR@loadbefore{fontenc}

```

File 60 **lwarp-fontspec.sty**

§ 136 Package **fontspec**

Pkg **fontspec** Error if fontspec is loaded after lwarp.

Discard all options for lwarp-fontspec:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{fontspec}
2 \LWR@loadbefore{fontspec}
```

File 61 **lwarp-footmisc.sty**

§ 137 Package **footmisc**

(Based on original code by ROBIN FAIRBAIRNS.)

Pkg **footmisc** footmisc is emulated during HTML output, and the footmisc package is ignored.

```
1 \LWR@ProvidesPackageDrop{footmisc}
```

Some nullified commands:

```
2 \newcommand{\footnotelayout}{}
3 \newcommand{\setfnsymbol}[1]{}
4 \NewDocumentCommand{\DefineFNSymbols}{s m o m}{}
5
6 \newdimen\footnotemargin
7 \footnotemargin1.8em\relax
8
9 \newcommand*\hangfootparskip{0.5\baselineskip}
10 \newcommand*\hangfootparindent{0em}%
11
12 \let\pagefootnoterule\footnoterule
13 \let\mpfootnoterule\footnoterule
14 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
15
16 \providecommand*\multiplefootnotemarker}{3sp}
17 \providecommand*\multfootsep}{,}
```

Using cleveref:

```
18 \providecommand*\footref}[1]{\labelcref{#1}}
```

The following work as-is:

```
19 \newcommand\mpfootnotemark{%
20   \@ifnextchar[%
21     \@xmpfootnotemark
22     {%
23       \stepcounter\@mpfn
24       \protected@xdef\@thefnmark{\thempfn}%
25       \@footnotemark
26     }%
27 }
28 \def\@xmpfootnotemark[#1]{%
29   \begingroup
30     \csname c@\@mpfn\endcsname #1\relax
31     \unrestored@protected@xdef\@thefnmark{\thempfn}%
32   \endgroup
33   \@footnotemark
34 }
```

File 62 `lwarp-footnote.sty`

§ 138 Package **footnote**

Pkg `footnote` `footnote` is used with minor patches.

for HTML output: `1 \LWR@ProvidesPackagePass{footnote}`

Removed print-version formatting:

```
2 \def\fn@startnote{%
3   \@parboxrestore%
4   \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
5   \color@begingroup% *** conflicts with lwarp
6 }
7
8 \let\fn@endnote\color@endgroup% *** conflicts with lwarp
9 \newcommand*\fn@endnote}{%
10 \LWR@htmltagc{/\LWR@tagregularparagraph}%
11 \LWR@orignewline%
12 }
```

Removed print-version formatting:

```
13 \def\fn@startfntext{%
14   \setbox\z@\vbox\bgroup%
```

```

15 \fn@startnote%
16 \fn@prefntext%
17 \ignorespaces%
18 }

```

Removed print-version formatting, added closing paragraph tag:

```

19 \def\fn@endfntext{%
20   \LWR@htmltagc{/\LWR@tagregularparagraph}%
21   \LWR@orignewline%
22   \fn@postfntext%
23   \egroup%
24   \beginngroup%
25   \let\@makefntext\@empty%
26   \let\@finalstrut\@gobble%
27   \LetLtxMacro\rule\@gobbletwo% *8* also the optional argument?
28   \@footnotetext{\unvbox\z@}%
29   \endgroup%
30 }

```

These have been redefined, so re-\let them again:

```

31 \let\endfootnote\fn@endfntext
32 \let\endfootnotetext\endfootnote

```

File 63 **lwarp-footnotehyper.sty**

§ 139 Package **footnotehyper**

Pkg footnotehyper footnotehyper is a hyperref-safe version of footnote. For lwarp, footnotehyper is emulated.

for HTML output: Discard all options for lwarp-footnotehyper:

```

1 \RequirePackage{footnote}
2 \LWR@ProvidesPackageDrop{footnotehyper}

```

File 64 **lwarp-framed.sty**

§ 140 Package **framed**

(Based on original code by DONALD ARSENEAU.)

Pkg framed framed is supported and patched by lwarp.

for HTML output: Accept all options for lwarp-framed:

```

1 \LWR@ProvidesPackagePass{framed}
2 \RequirePackage{xcolor}% for \convertcolorspec

3
4 \renewenvironment{framed}{%
5 \LWR@forcenewpage
6 \BlockClass{framed}%
7 }
8 {\endBlockClass}
9
10 \renewenvironment{oframed}{%
11 \LWR@forcenewpage
12 \BlockClass{framed}%
13 }
14 {\endBlockClass}
15
16
17 \renewenvironment{shaded}{%
18 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
19 \LWR@forcenewpage
20 \BlockClass[background: \#\LWR@tempcolor]{framed}%
21 }
22 {\endBlockClass}
23
24 \renewenvironment{shaded*}{%
25 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
26 \LWR@forcenewpage
27 \BlockClass[background: \#\LWR@tempcolor]{framed}%
28 }
29 {\endBlockClass}
30
31
32 \renewenvironment{leftbar}{%
33 \LWR@forcenewpage
34   \BlockClass{framedleftbar}
35   \def\FrameCommand{}%
36   \MakeFramed {}
37 }%
38 {\endMakeFramed\endBlockClass}
39
40
41 \renewenvironment{snugshade}{%
42 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
43 \LWR@forcenewpage
44 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
45 }
46 {\endBlockClass}
47

```

```

48 \renewenvironment{snugshade*}{%
49 \convertcolorspec{named}{shadecolor}{HTML}\LWR@tempcolor%
50 \LWR@forcenewpage
51 \BlockClass[background: \#\LWR@tempcolor]{snugframed}%
52 }
53 {\endBlockClass}
54
55 \let\oframed\framed
56 \let\endoframed\endframed
57
58
59 \RenewEnviron{titled-frame}[1]{%
60 \CustomFBox{#1}{}{Opt}{Opt}{Opt}{Opt}{\BODY}
61 }

\CustomFBox {<toptitle>} {<bottitle>} {<thicknesstop>} {<bottom>} {<left>} {<right>}
{<text contents>}

62 \renewcommand{\CustomFBox}[7]{%
63 \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
64 \LWR@forcenewpage
65 \begin{BlockClass}[border: 3px solid \#\LWR@tempcolor]{framed}%
66 \ifthenelse{\isempty{#1}}{}{% not empty
67   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
68     \textcolor{TFTitleColor}{\textbf{#1}}%
69   \end{BlockClass}
70 }% not empty
71
72 #7
73
74 \ifthenelse{\isempty{#2}}{}{% not empty
75   \convertcolorspec{named}{TFFrameColor}{HTML}\LWR@tempcolor%
76   \begin{BlockClass}[background: \#\LWR@tempcolor]{framedtitle}%
77     \textcolor{TFTitleColor}{\textbf{#2}}%
78   \end{BlockClass}
79 }% not empty
80 \end{BlockClass}
81 }

\TitleBarFrame [<marker>] {<title>} {<contents>}

82 \renewcommand\TitleBarFrame[3][ ]{
83 \CustomFBox
84   {#2}{}%
85   \fboxrule\fboxrule\fboxrule\fboxrule
86   {#3}%
87 }

88 \renewcommand{\TF@Title}[1]{#1}

```

MakeFramed $\{\langle settings \rangle\}$

```
89 \let\MakeFramed\relax
90 \let\endMakeFramed\relax
91
92 \NewEnviron{MakeFramed}[1]{%
93 \FrameCommand{\begin{minipage}{\linewidth}\BODY\end{minipage}}%
94 }
```

$\fb@put@frame$ $\{\langle frame\ cmd\ no\ split \rangle\}$ $\{\langle frame\ cmd\ split \rangle\}$

```
95 \renewcommand*\fb@put@frame}[2]{%
96 \relax%
97 \@tempboxa%
98 }
```

File 65 **lwarp-ftnright.sty**

§ 141 Package **ftnright**

Pkg `ftnright` ftnright is ignored.

for HTML output: Discard all options for lwarp-ftnright:

```
1 \LWR@ProvidesPackageDrop{ftnright}
```

File 66 **lwarp-fullpage.sty**

§ 142 Package **fullpage**

Pkg `fullpage` Emulated.

for HTML output: Discard all options for lwarp-fullpage:

```
1 \LWR@ProvidesPackageDrop{fullpage}
```

File 67 **lwarp-fullwidth.sty**

§ 143 Package **fullwidth**

Pkg `fullwidth` fullwidth is emulated.

A minipage is used, of no HTML width.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{fullwidth}
2 \newenvironment*{fullwidth}[1][1]{}
3 \minipagefullwidth%
4 \minipage{\linewidth}%
5 }
6 {}
7 \endminipage%
8 }
```

File 68 **lwarp-geometry.sty**

§ 144 Package **geometry**

Pkg **geometry** geometry is preloaded by lwarp, but must be nullified as seen by the user's source code.

for HTML output: Discard all options for lwarp-geometry:

```

1 \LWR@ProvidesPackageDrop{geometry}
2 \renewcommand*{\geometry}[1]{}
3 \renewcommand*{\newgeometry}[1]{}
4 \renewcommand*{\restoregeometry}{}
5 \renewcommand*{\savegeometry}[1]{}
6 \renewcommand*{\loadgeometry}[1]{}

```

File 69 **lwarp-glossaries.sty**

§ 145 Package **glossaries**

Pkg **glossaries** xindy is required for glossaries.

The default `style=item` option for glossaries conflicts with lwarp, so the style is forced to `index` instead.

The page number list in the printed form would become `\namerefs` in HTML, which could become a very long string if many items are referenced. For now, the number list is simply turned off.

[placement and toc options](#) The glossaries may be placed in a numbered or unnumbered section, given a toc entry, and placed inline or on their own HTMLpage:

Numbered section, on its own HTML page:

```
\usepackage[xindy,toc,numberedsection=nolabel]{glossaries}
...
\printglossaries
```

Unnumbered section, inline with the current HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\printglossaries
```

Unnumbered section, on its own HTML page:

```
\usepackage[xindy,toc]{glossaries}
...
\ForceHTMLPage
\printglossaries
```

Opt `IndexLanguage` The `lwarp` package takes an option `IndexLanguage=english` to set the language used by `xindy`. This is passed to `xindy` using its `-L` option, and is used for both index and glossary generation.

Opt `lwarpmk printglossary` `lwarpmk` has the commands `lwarpmk printglossary` and `lwarpmk htmlglossary` to process the glossaries created by `glossaries` using `xindy`.

Opt `lwarpmk htmlglossary`

for HTML output:

```
1 \PassOptionsToPackage{xindy}{glossaries}
2 \LWR@ProvidesPackagePass{glossaries}
3 \setupglossaries{nonumberlist}
4 \setglossarystyle{index}
```

Patched to fix TOC pointing to the previous page:

```
5 \renewcommand*{\@p@glossarysection}[2]{%
6   \glsclearpage
7   \phantomsection
8   \ifdefempty\@glossarysecstar
9   {%
10    \csname\@glossarysec\endcsname{#2}%
11  }%
12  {%
```

In the original, the TOC entry was made before the section, thus linking to the `phantomsection` in the printed version, but for HTML this caused the link to point to the page before the glossaries. Here, the TOC entry is made after the section is created:

```
13   \csname\@glossarysec\endcsname*{#2}%
14   \gls@toc{#1}{\@glossarysec}% Moved after the previous line.
15  }%
16  \@glossaryseclabel
17 }
```

File 70 `lwarp-graphics.sty`

§ 146 Package **graphics**

Pkg `graphics` `graphics` is emulated.

for HTML output: `1 \LWR@ProvidesPackagePass{graphics}`

§ 146.1 **Graphics extensions**

`\DeclareGraphicsExtensions` `{\list}`

`\AtBeginDocument` allow SVG files instead of PDF:

```
2 \AtBeginDocument{
3 \DeclareGraphicsExtensions{.svg,.SVG,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}
4 \DeclareGraphicsRule{.svg}{svg}{.svg}{}
5 \DeclareGraphicsRule{.SVG}{svg}{.SVG}{}
6 }
```

Inside a `lateximage`, allow PDF instead of SVG:

```
7 \appto\LWR@restoreorigformatting{%
8 \DeclareGraphicsExtensions{.pdf,.PDF,.gif,.GIF,.png,.PNG,.jpg,.JPG,.jpeg,.JPEG}%
9 }
```

§ 146.2 Length conversions and graphics options



whitespace

A scaled image in \TeX by default takes only as much space on the page as it requires, but HTML browsers use as much space as the original unscaled image would have taken, with the scaled image over- or under-flowing the area.

```
10 \AtBeginDocument{
11 \renewcommand*\unitspace}{ }
12 }
```

Used to store the user's selected dimensions and HTML class.

The class defaults to “inlineimage” unless changed by a `class=xyx` option.

```
13 \newlength{\LWR@igwidth}
14 \newlength{\LWR@igheight}
15 \newcommand*\LWR@igwidthstyle{}
16 \newcommand*\LWR@igheightstyle{}
17 \newcommand*\LWR@igorigin{}
18 \newcommand*\LWR@igangle{}
19 \newcommand*\LWR@igxscale{1}
20 \newcommand*\LWR@igyscale{1}
21 \newcommand*\LWR@igclass{inlineimage}
```

Set the actions of each of the key/value combinations for `\includegraphics`. Many are ignored.

If an optional width was given, set an HTML style:

```
22 \define@key{igraph}{width}{%
23 \setlength{\LWR@igwidth}{#1}%
24 \ifthenelse{\lengthtest{\LWR@igwidth > 0pt}}{%
25 {%
```

Default to use the converted fixed length given:

```
26 \uselengthunit{PT}%
27 \renewcommand*\LWR@igwidthstyle}{width:\rndprintlength{\LWR@igwidth}}%
```

If ex or em dimensions were given, use those instead:

```
28 \IfEndWith{#1}{ex}%
29 {\renewcommand*\LWR@igwidthstyle}{width:#1}}% yes ex
30 {}% not ex
31 \IfEndWith{#1}{em}%
32 {\renewcommand*\LWR@igwidthstyle}{width:#1}}% yes em
33 {}% not em
34 \IfEndWith{#1}{\}%
35 {\renewcommand*\LWR@igwidthstyle}{width:#1}}% yes percent
```

```

36   {}% not percent
37   \IfEndWith{#1}{px}%
38   {\renewcommand*{\LWR@igwidthstyle}{width:#1}}% yes px
39   {}% not px
40 }{}% end of length > Opt
41 }

```

If an optional height was given, set an HTML style:

```

42 \define@key{igraph}{height}{%
43 \setlength{\LWR@igheight}{#1}%
44 \ifthenelse{\lengthtest{\LWR@igheight > Opt}}%
45 {%

```

Default to use the converted fixed length given:

```

46   \uselengthunit{PT}%
47   \renewcommand*{\LWR@igheightstyle}{%
48   height:\rndprintlength{\LWR@igheight} %
49   }%

```

If ex or em dimensions were given, use those instead:

```

50   \IfEndWith{#1}{ex}%
51   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes ex
52   {}% not ex
53   \IfEndWith{#1}{em}%
54   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes em
55   {}% not em
56   \IfEndWith{#1}{\}%
57   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes percent
58   {}% not percent
59   \IfEndWith{#1}{px}%
60   {\renewcommand*{\LWR@igheightstyle}{height:#1}}% yes px
61   {}% not px
62 }{}% end of length > Opt
63 }

```

Handle origin key:

```

64 \define@key{igraph}{origin}{%
65 \renewcommand*{\LWR@igorigin}{#1}%
66 }

```

Handle angle key:

```

67 \define@key{igraph}{angle}{\renewcommand*{\LWR@igangle}{#1}}

```

Handle class key:

```
68 \define@key{igraph}{class}{\renewcommand*\LWR@igclass}{#1}}
69
```

It appears that `graphicx` does not have separate keys for `xscale` and `yscale`. `scale` adjusts both at the same time.

```
70 \define@key{igraph}{scale}{%
71 \renewcommand*\LWR@igxscale}{#1}%
72 \renewcommand*\LWR@igyscale}{#1}}
```

Numerous ignored keys:

```
73 \define@key{igraph}{bb}{}
74 \define@key{igraph}{bbllx}{}
75 \define@key{igraph}{bbllx}{}
76 \define@key{igraph}{bburx}{}
77 \define@key{igraph}{bbury}{}
78 \define@key{igraph}{natwidth}{}
79 \define@key{igraph}{natheight}{}
80 \define@key{igraph}{hiresbb}{}
81 \define@key{igraph}{viewport}{}
82 \define@key{igraph}{trim}{}
83 \define@key{igraph}{totalheight}{}
84 \define@key{igraph}{keepaspectratio}{}
85 \define@key{igraph}{clip}{}
86 \define@key{igraph}{draft}{}
87 \define@key{igraph}{type}{}
88 \define@key{igraph}{ext}{}
89 \define@key{igraph}{read}{}
90 \define@key{igraph}{command}{}

```

`\LWR@rotstyle` $\{\langle prefix \rangle\} \{\langle degrees \rangle\}$

Prints the rotate style with the given prefix.

`prefix` is `-ms-` or `-webkit-` or nothing, and is used to generate three versions of the `transform:rotate` style.

```
91 \newcommand*\LWR@rotstyle}[2]{%
92 #1transform:rotate(-#2deg);
93 }
```

`\LWR@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 }
```

§ 146.3 `\includegraphics`

Bool `LWR@infloatrow` Used to compute `\linewidth`.

```
97 \newbool{LWR@infloatrow}
98 \boolfalse{LWR@infloatrow}
```

`\LWR@opacity` may be set by the transparent package. For HTML it is only used for `\includegraphics`.

```
99 \def\LWR@opacity{1}
```

`\LWR@includegraphicsb` * [`<2: options>`] [`<3: options>`] [`<4: filename>`]

graphics syntax is `\includegraphics * [<llx, lly>] [<urx, ury>] [<file>]`

graphicx syntax is `\includegraphics [<key values>] [<file>]`

If #3 is empty, only one optional argument was given, thus graphicx syntax.

```
100 \NewDocumentCommand{\LWR@includegraphicsb}{s o o m}
101 {%
102 \LWR@traceinfo{LWR@includegraphicsb #4}%
```

Start the image tag on a new line, allow PDF output word wrap:

```
103 \LWR@origtilde \LWR@orignewline%
```

Temporarily compute `\linewidth`, `\textwidth`, `\textheight` arguments with a 6x9 inch size until the next `\endgroup`.

```
104 \ifthenelse{\cinttest{\value{LWR@minipagedepth}}{=}{0}}{%
105 {%
106   \ifbool{LWR@infloatrow}%
107   {}
108   {% not in a minipage or a floatrow:
109     \setlength{\linewidth}{6in}%
110     \setlength{\textwidth}{6in}%
111     \setlength{\textheight}{9in}%
112   }%
113 }{}}%
```

```

114 \renewcommand*\Gin@setfile}[3]{%
115 \LWR@traceinfo{Gin@setfile ##3}%
116 \xdef\LWR@parsedfilename{##3}%
117 }%
118 \Gin@include@graphics{\detokenize{#4}}%
119 \filename@parse{\LWR@parsedfilename}%

```

For correct em sizing during the width and height conversions:

```
120 \large%
```

Reset some defaults, possibly will be changed below if options were given:

```

121 \setlength{\LWR@igwidth}{Opt}%
122 \setlength{\LWR@igheight}{Opt}%
123 \renewcommand*\LWR@igwidthstyle{}%
124 \renewcommand*\LWR@igheightstyle{}%
125 \renewcommand*\LWR@igorigin{}%
126 \renewcommand*\LWR@igangle{}%
127 \renewcommand*\LWR@igxscale}{1}%
128 \renewcommand*\LWR@igyscale}{1}%
129 \renewcommand*\LWR@igclass}{inlineimage}%

```

If #3 is empty, only one optional argument was given, thus graphicx syntax:

```

130 \IfValueF{#3}{%
131 \IfValueTF{#2}%
132 {\setkeys{igraph}{#2}}%
133 {\setkeys{igraph}{}}%
134 }%

```

Create the HTML reference with the graphicspath, filename, extension, alt tag, style, and class.

The `\LWR@origtilde` adds space between tags in case this is being done inside a `\savebox` where `\newline` has no effect.

```

135 \LWR@traceinfo{LWR@includegraphicsb: about to create href}%
136 \href{\LWR@parsedfilename}%
137 {% start of href
138 \LWR@traceinfo{LWR@includegraphicsb: about to LWR@htmltag}%
139 \LWR@htmltag{% start of image tags
140 img src="\LWR@parsedfilename" \LWR@orignewline%
141 \LWR@origtilde{} alt="[filename@base]" \LWR@orignewline%

```

Only include a style tag if a width, height, angle, or scale was given:

```
142 \ifthenelse{
```

```

143 \NOT\equal{\LWR@igwidthstyle}{ } \OR
144 \NOT\equal{\LWR@igheightstyle}{ } \OR
145 \NOT\equal{\LWR@igorigin}{ } \OR
146 \NOT\equal{\LWR@igangle}{ } \OR
147 \NOT\equal{\LWR@igxscale}{1} \OR
148 \NOT\equal{\LWR@igyscale}{1}
149 }%
150 {\LWR@origtilde{ } style="%
151 \ifthenelse{\NOT\equal{\LWR@igwidthstyle}{ }}%
152 {\LWR@igwidthstyle; }}%
153 \ifthenelse{\NOT\equal{\LWR@igheightstyle}{ }}%
154 {\LWR@igheightstyle; }}%
155 \ifthenelse{\NOT\equal{\LWR@igorigin}{ }}%
156 {\LWR@origtilde{ } transform-origin: \LWR@originnames{\LWR@igorigin}; \LWR@originewline}{ }%
157 \ifthenelse{\NOT\equal{\LWR@igangle}{ }}%
158 {%
159 \LWR@rotstyle{-ms-}{\LWR@igangle}%
160 \LWR@rotstyle{-webkit-}{\LWR@igangle}%
161 \LWR@rotstyle{ }{\LWR@igangle%
162 } }%
163 \ifthenelse{\NOT\equal{\LWR@igxscale}{1}\OR%
164 \NOT\equal{\LWR@igyscale}{1}}%
165 {\LWR@scalestyle{-ms-}{\LWR@igxscale}{\LWR@igyscale}%
166 \LWR@scalestyle{-webkit-}{\LWR@igxscale}{\LWR@igyscale}%
167 \LWR@scalestyle{ }{\LWR@igxscale}{\LWR@igyscale}}}%
168 %
169 \ifthenelse{\NOT\equal{\LWR@opacity}{1}}%
170 {opacity: \LWR@opacity; }%
171 { }%
172 %
173 " \LWR@originewline}{ }%

```

Set the class:

```

174 \LWR@origtilde{ } class="\LWR@igclass" \LWR@originewline%
175 }% end of image tags
176 }% end of href
177 \endgroup

```

Return to small-sized output:

```

178 \LWR@origscriptsize
179 \LWR@traceinfo{\LWR@includegraphicsb done}%
180 }

```

`\includegraphics` [*{key=val}*] *{filename}*

Handles width and height, converted to fixed width and heights.

Converts any .pdf references to .svg for HTML

The user should always refer to .pdf in the document source.

```
181 \LetLtxMacro\LWR@originincludegraphics\includegraphics
182
183 \AtBeginDocument{
184 \renewcommand*\includegraphics{
185 {%
```

This graphic should trigger an HTML paragraph even if alone, so ensure that are doing paragraph handling:

```
186 \LWR@traceinfo{includegraphics}
187 \LWR@ensuredoingapar%
188 \begingroup%
189 \catcode'\_ =12%
190 \LWR@includegraphicsb%
191 }% includegraphics
192 }% AtBeginDocument
```

`\LWR@rotboxorigin` Holds the origin key letters.

```
193 \newcommand*\LWR@rotboxorigin{-}
```

`\LWR@originname` $\{\langle letter \rangle\}$

Given one \TeX origin key value, translate into an HTML origin word:

```
194 \newcommand*\LWR@originname}[1]{%
195 \ifthenelse{\equal{#1}{t}}{top}{}%
196 \ifthenelse{\equal{#1}{b}}{bottom}{}%
197 \ifthenelse{\equal{#1}{c}}{center}{}%
198 \ifthenelse{\equal{#1}{l}}{left}{}%
199 \ifthenelse{\equal{#1}{r}}{right}{}%
200 }
```

`\LWR@originnames` $\{\langle letters \rangle\}$

Given one- or two-letter \TeX origin key values, translate into HTML origin words:

```
201 \newcommand*\LWR@originnames}[1]{%
202 \StrChar{#1}{1}[\LWR@strresult]%
203 \LWR@originname{\LWR@strresult}%
204 \StrChar{#1}{2}[\LWR@strresult]%
205 \LWR@originname{\LWR@strresult}%
206 }
```

Handle the origin key for `\rotatebox`:

```
207 \define@key{krotbox}{origin}{%
208 \renewcommand*\LWR@rotboxorigin}{#1}%
209 }
```

These keys are ignored:

```
210 \define@key{krotbox}{x}{}
211 \define@key{krotbox}{y}{}
212 \define@key{krotbox}{units}{}

```

`\rotatebox` [*keyval list*] {*angle*} {*text*}

```
213 \LetLtxMacro\LWR@origrotatebox\rotatebox
214
215 \AtBeginDocument{
216 \RenewDocumentCommand{\rotatebox}{O{} m +m}{%

```

Reset the origin to “none-given”:

```
217 \renewcommand*\LWR@rotboxorigin}{}

```

Process the optional keys, which may set `\LWR@rotateboxorigin`:

```
218 \setkeys{krotbox}{#1}%

```

Select inline-block so that HTML will transform this span:

```
219 \LWR@htmltagc{span style="display: inline-block; %

```

If an origin was given, translate and print the origin information:

```
220 \ifthenelse{\NOT\equal{\LWR@rotboxorigin}{} }%
221 {transform-origin: \LWR@originnames{\LWR@rotboxorigin};\LWR@origtilde}{}%

```

Print the rotation information:

```
222 \LWR@rotstyle{-ms-}{#2} %
223 \LWR@rotstyle{-webkit-}{#2} %
224 \LWR@rotstyle}{#2} %
225 "{}\LWR@orignewline%

```

Print the text to be rotated:

```
226 \begin{LWR@nestspan}%
227 #3%

```

Close the span:

```
228 \LWR@htmltagc{/span}%
229 \end{LWR@nestspan}%
230 }
231 }% AtBeginDocument
```

`\scalebox` $\langle h-scale \rangle$ $\langle v-scale \rangle$ $\langle text \rangle$

```
232 \LetLtxMacro\LWR@origscalebox\scalebox
233
234 \AtBeginDocument{
235 \RenewDocumentCommand{\scalebox}{m o m}{%
```

Select inline-block so that HTML will transform this span:

```
236 \LWR@htmltagc{span style="display: inline-block; %
```

Print the scaling information:

```
237 \LWR@scalestyle{-ms-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
238 \LWR@scalestyle{-webkit-}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
239 \LWR@scalestyle{}{#1}{\IfNoValueTF{#2}{#1}{#2}} %
240 "{}}%
```

Print the text to be scaled:

```
241 \begin{LWR@nestspan}%
242 #3%
```

Close the span:

```
243 \LWR@htmltagc{/span}%
244 \end{LWR@nestspan}%
245 }
246 }% AtBeginDocument
```

`\reflectbox` $\langle text \rangle$

```
247 \let\LWR@origreflectbox\reflectbox
248
249 \AtBeginDocument{
250 \renewcommand{\reflectbox}[1]{\scalebox{-1}[1]{#1}}
251 }
```

`\resizebox` $\langle h-length \rangle$ $\langle v-length \rangle$ $\langle text \rangle$

Simply prints its text argument.

```
252 \LetLtxMacro\LWR@origresizebox\resizebox
253
254 \AtBeginDocument{
255 \renewcommand{\resizebox}[3]{#3}
256 }
```

File 71 **lwarp-graphics.sty**

§ 147 Package **graphics**

Pkg **graphics** graphics is emulated.

graphics 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{graphics}`

File 72 **lwarp-grffile.sty**

§ 148 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 73 **lwarp-hyperref.sty**

§ 149 Package **hyperref**

Pkg **hyperref** hyperref is emulated during HTML output, and the hyperref package is ignored.

for HTML output:

```

1% \LWR@ProvidesPackageDrop{hyperref}
2\typeout{Using the lwarp html version of package 'hyperref' -- discarding options.}
3\typeout{   Are not using ProvidesPackage, so that other packages}
4\typeout{   do not attempt to patch lwarp's version of 'hyperref'.}
5% \ProvidesPackage{lwarp-#1-#2}
6\DeclareOption*{}
7\ProcessOptions\relax

8\newcommand*\hypersetup}[1]{}
9\newcommand*\hyperbaseurl}[1]{}

\hyperimage  {<url>} {<alt text>}
Insert an image with alt text:

10\NewDocumentCommand{\LWR@hyperimageb}{m +m}{%
11\LWR@htmltag{img src="#1" alt="#2" class="hyperimage"{}%}
12\endgroup%
13\LWR@ensuredoingapar%
14}
15
16\newcommand{\hyperimage}{%
17\LWR@ensuredoingapar%
18\begingroup\catcode'\_ =12
19\LWR@hyperimageb%
20}

\hyperdef  {<1: category>} {<2: name>} {<3: text>}
Creates an HTML anchor to category.name with the given text.

21\NewDocumentCommand{\hyperdef}{m m +m}{%
22\LWR@ensuredoingapar%
23\LWR@subsublabel{#1.#2}%
24#3%
25}

\LWR@hyperrefb  {<1: URL>} {<2: category>} {<3: name>} {<4: text>}
Creates an HTML link to URL#category.name with the given text.

26\NewDocumentCommand{\LWR@hyperrefb}{m m m +m}{%
27\LWR@htmltag{a href="#1\LWR@hashmark#2.#3"%}
28#4%
29\LWR@htmltag{/a}%
30\endgroup%
31}

\LWR@hyperrefc  [<label>] {<text>}

```

Creates text as an HTML link to the \LaTeX label.

```
32 \NewDocumentCommand{\LWR@hyperrefc}{0{label} +m}{
33 \LWR@startref{#1}%
34 #2%
35 \LWR@htmltag{/a}%
36 \endgroup%
37 }
```

`\hyperref` $\langle 1: URL \rangle \langle 2: category \rangle \langle 3: name \rangle \langle 4: text \rangle$ — or —
 $[\langle 1: label \rangle] \langle 2: text \rangle$

```
38 \newcommand{\hyperref}{%
39 \LWR@ensuredoingapar%
40 \begingroup\catcode'\_ =12
41 \@ifnextchar[\LWR@hyperrefc\LWR@hyperrefb%
42 }
```

`\hypertarget` $\langle name \rangle \langle text \rangle$

Creates an anchor to name with the given text.

```
43 \NewDocumentCommand{\hypertarget}{m +m}{%
44 \label{#1}%
45 #2%
46 }
```

`\hyperlink` $\langle name \rangle \langle text \rangle$

Creates a link to the anchor created by `hypertarget`, with the given link text.

```
47 \NewDocumentCommand{\hyperlink}{m +m}{%
48 \hyperref[ #1 ]{ #2 }%
49 }
```

`\autoref` * $\langle label \rangle$

For HTML, `\cleveref` is used instead.

```
50 \NewDocumentCommand{\autoref}{s m}{%
51 \IfBooleanTF{#1}{\ref{#2}}{\cref{#2}}%
52 }
```

`\autopageref` $\langle label \rangle$

For HTML, `\cleveref` is used instead.

```
53 \NewDocumentCommand{\autopageref}{s m}{%
54 \IfBooleanTF{#1}{\cpageref{#2}}{\cref{#2}}%
55 }
```

`\pdfstringdef` $\{\langle macroname \rangle\} \{\langle T_{E}Xstring \rangle\}$
56 `\newcommand{\pdfstringdef}[2]{}`

`\pdfbookmark` $[\langle level \rangle] \{\langle text \rangle\} \{\langle name \rangle\}$
57 `\newcommand{\pdfbookmark}[3] [] {}`

`\currentpdfbookmark` $\{\langle text \rangle\} \{\langle name \rangle\}$
58 `\newcommand{\currentpdfbookmark}[2]{}`

`\subpdfbookmark` $\{\langle text \rangle\} \{\langle name \rangle\}$
59 `\newcommand{\subpdfbookmark}[2]{}`

`\belowpdfbookmark` $\{\langle text \rangle\} \{\langle name \rangle\}$
60 `\newcommand{\belowpdfbookmark}[2]{}`

`\texorpdfstring` $\{\langle T_{E}Xstring \rangle\} \{\langle PDFstring \rangle\}$
61 `\newcommand{\texorpdfstring}[2]{#2}`

`\hypercalcbp` $\{\langle dimen \rangle\}$ From hyperref.
62 `\def\hypercalcbp#1{%`
63 `\strip@pt\dimexpr 0.99626401\dimexpr(#1)\relax\relax`
64 `}%`

`\Acrobatmenu` $\{\langle menuoption \rangle\} \{\langle text \rangle\}$
65 `\newcommand{\Acrobatmenu}[2]{}`

`\TextField` $[\langle parameters \rangle] \{\langle label \rangle\}$
66 `\newcommand*{\TextField}[2] [] {}`

`\CheckBox` $[\langle parameters \rangle] \{\langle label \rangle\}$
67 `\newcommand*{\CheckBox}[2] [] {}`

`\ChoiceMenu` $[\langle parameters \rangle] \{\langle label \rangle\} \{\langle choices \rangle\}$
68 `\newcommand{\ChoiceMenu}[3] [] {}`

`\PushButton` [*<parameters>*] {*<label>*}

69 `\newcommand*\PushButton}[2] [] {}`

`\Submit` [*<parameters>*] {*<label>*}

70 `\newcommand*\Submit}[2] [] {}`

`\Reset` [*<parameters>*] {*<label>*}

71 `\newcommand*\Reset}[2] [] {}`

`\LayoutTextField` {*<label>*} {*<field>*}

72 `\newcommand*\LayoutTextField}[2] {}`

`\LayoutChoiceField` {*<label>*} {*<field>*}

73 `\newcommand*\LayoutChoiceField}[2] {}`

`\LayoutCheckField` {*<label>*} {*<field>*}

74 `\newcommand*\LayoutCheckField}[2] {}`

`\MakeRadioField` {*<width>*} {*<height>*}

75 `\newcommand*\MakeRadioField}[2] {}`

`\MakeCheckField` {*<width>*} {*<height>*}

76 `\newcommand*\MakeCheckField}[2] {}`

`\MakeTextField` {*<width>*} {*<height>*}

77 `\newcommand*\MakeTextField}[2] {}`

`\MakeChoiceField` {*<width>*} {*<height>*}

78 `\newcommand*\MakeChoiceField}[2] {}`

`\MakeFieldButton` {*<text>*}

79 `\newcommand\MakeFieldButton}[1] {}`

File 74 `lwarp-hyperxmp.sty`

§ 150 Package **hyperxmp**

Pkg hyperxmp Emulated.

for HTML output: Discard all options for lwarp-hyperxmp:

```
1 \LWR@ProvidesPackageDrop{hyperxmp}
```

File 75 `lwarp-idxlayout.sty`

§ 151 Package **idxlayout**

Pkg idxlayout Emulated.

for HTML output: Discard all options for lwarp-idxlayout:

```
1 \LWR@ProvidesPackageDrop{idxlayout}

2 \newcommand{\LWR@indexprenote}{}
3
4 \renewcommand*\printindex
5 {
6 \LWR@startpars
7
8 \LWR@indexprenote
9
10 \LWR@origprintindex
11 }
12
13 \newcommand{\setindexprenote}[1]{\renewcommand{\LWR@indexprenote}{#1}}
14 \newcommand*\noindexprenote{\renewcommand{\LWR@indexprenote}{} }
15
16 \newcommand{\idxlayout}[1]{}
17 \newcommand*\indexfont{}
18 \newcommand*\indexjustific{}
19 \newcommand*\indexsubsdelim{}
20 \newcommand*\indexstheadcase{}
```

File 76 `lwarp-indentfirst.sty`

§ 152 Package **indentfirst**

Pkg `indentfirst` `indentfirst` is ignored.

Discard all options for `lwarp-indentfirst`:

for HTML output: `1 \LWR@ProvidesPackageDrop{indentfirst}`

File 77 `lwarp-inputenc.sty`

§ 153 Package **inputenc**

Pkg `inputenc` Error if `inputenc` is loaded after `lwarp`.

Discard all options for `lwarp-inputenc`:

for HTML output: `1 \LWR@ProvidesPackageDrop{inputenc}`

`2 \LWR@loadbefore{inputenc}`

File 78 `lwarp-keyfloat.sty`

§ 154 Package **keyfloat**

Pkg `keyfloat` `keyfloat` is supported with minor adjustments.

for HTML output: `1 \LWR@ProvidesPackagePass{keyfloat}`

After `keyfloat` has loaded:

`2 \AtBeginDocument{`

`3 \let\KFLT@boxinner\relax`

`4 \let\endKFLT@boxinner\relax`

`5`

`6 \NewEnviron{KFLT@boxinner}`

`7 {%`

```

8 \LWR@traceinfo{kflt@boxinner}%
9 \LWR@stoppars%
10 \KFLT@frame{\BODY}%
11 \LWR@startpars%
12 \LWR@traceinfo{ended kflt@boxinner}%
13 }

14 \DeclareDocumentEnvironment{KFLT@marginfloat}{0{-1.2ex} m}
15 {% start
16 \LWR@maybeinthisfloat%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
20 \LWR@startpars%
21 \captionsetup{type=#2}%
22 }
23 {
24 \LWR@htmldivclassend{div}
25 }

26 \DeclareDocumentEnvironment{marginfigure}{o}
27 {\begin{KFLT@marginfloat}{figure}}
28 {\end{KFLT@marginfloat}}
29

30 \DeclareDocumentEnvironment{margintable}{o}
31 {\begin{KFLT@marginfloat}{table}}
32 {\end{KFLT@marginfloat}}

33 \DeclareDocumentEnvironment{keywrap}{m +m}
34 {%
35 \begin{BlockClass}{marginblock}
36 \setlength{\linewidth}{#1}
37 #2%
38 \end{BlockClass}
39 }
40 {%
41 }

42 }% AtBeginDocument

```

File 79 **lwarp-layout.sty**

§ 155 Package **layout**

Pkg layout layout is ignored.

for HTML output: Discard all options for lwarp-layout:

```
1 \LWR@ProvidesPackageDrop{layout}
2 \NewDocumentCommand{\layout}{s}{}
```

File 80 **lwarp-letterspace.sty**

§ 156 Package **letterspace**

Pkg letterspace letterspace is a subset of microtype, which is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-letterspace:

```
1 \LWR@ProvidesPackageDrop{letterspace}
2 \newcommand*\lsstyle{}
3 \newcommand\textls[2] []{}
4 \def\textls#1#{}
5 \newcommand*\lslig[1]{#1}
```

File 81 **lwarp-lettrine.sty**

§ 157 Package **lettrine**

(Based on original code by DANIEL FLIPO.)

Pkg lettrine Emulated.

for HTML output: Discard all options for lwarp-lettrine:

```
1 \LWR@ProvidesPackageDrop{lettrine}
```

The initial letter is in a of class lettrine, and the following text is in a of class lettrinertext. \lettrine [*keys*] {<letter>} {<additional text>}

```
2 \DeclareDocumentCommand{\lettrine}{o m m}{%
3 \InlineClass{lettrine}{#2}\InlineClass{lettrinertext}{#3} %
4 }
5
6 \newcounter{DefaultLines}
7 \setcounter{DefaultLines}{2}
8 \newcounter{DefaultDepth}
```

```

 9 \newcommand*\DefaultOptionsFile{\relax}
10 \newcommand*\DefaultLoversize{0}
11 \newcommand*\DefaultLraise{0}
12 \newcommand*\DefaultLhang{0}
13 \newdimen\DefaultFindent
14 \setlength{\DefaultFindent}{\z@}
15 \newdimen\DefaultNindent
16 \setlength{\DefaultNindent}{0.5em}
17 \newdimen\DefaultSlope
18 \setlength{\DefaultSlope}{\z@}
19 \newdimen\DiscardVskip
20 \setlength{\DiscardVskip}{0.2\p@}
21 \newif\ifLettrineImage
22 \newif\ifLettrineOnGrid
23 \newif\ifLettrineRealHeight
24
25 \newcommand*\LettrineTextFont{\scshape}
26 \newcommand*\LettrineFontHook{}
27 \newcommand*\LettrineFont[1]{\InlineClass{lettrine}{#1}}
28 \newcommand*\LettrineFontEPS[1]{\includegraphics[height=1.5ex]{#1}}

```

File 82 **lwarp-lips.sty**

§ 158 Package **lips**

Pkg lips lips is emulated during HTML output, and the lips package is ignored.

```

 1 % \LWR@ProvidesPackageDrop{lips}
 2 \PackageInfo{lwarp}{Using the lwarp version of package 'lips'.}%
 3 \ProvidesPackage{lwarp-lips}
 4
 5 \NewDocumentCommand{\Lips}{}{\textellipsis}
 6
 7 \NewDocumentCommand{\BracketedLips}{}{[\textellipsis]}
 8
 9 \let\lips\Lips
10 \let\olips\lips
11
12 \DeclareOption*{}
13 \DeclareOption{mla}{
14 \let\lips\BracketedLips
15 }
16 \ProcessOptions\relax
17
18 \newcommand \LPNobreakList {}

```

File 83 `lwarp-listings.sty`

§ 159 Package **listings**

(Based on original code by CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN.)

Pkg listings listings is supported with some limitations. Text formatting is not yet supported.

for HTML output: 1 `\begin{warpHTML}`

2 `\LWR@ProvidesPackagePass{listings}`

Patches to embed listings inside pre tags:

3 `\let\LWR@origlst@Init\lst@Init`

4 `\let\LWR@origlst@DeInit\lst@DeInit`

5

6 `\let\LWR@origlsthkEveryPar\lsthk@EveryPar`

7

8 `\renewcommand{\l@lstlisting}[2]{\hypertocfloat{1}{lstlisting}{lo1}{#1}{#2}}`

`\lst@Init` `{\backslash-processing}` Done at the start of a listing.

9 `\renewcommand{\lst@Init}[1]{%`

First, perform the listings initialization:

10 `\LWR@traceinfo{lst@Init}%`

11 `\renewcommand*{@capttype}{lstlisting}%`

12 `\LWR@origlst@Init{#1}%`

13 `\LWR@traceinfo{finished origlst@Init}%`

14 `\lst@ifdisplaystyle%`

Creating a display.

Disable line numbers, produce the `<pre>`, then reenable line numbers.

15 `\LWR@traceinfo{About to create verbatim.}%`

16 `\let\lsthk@EveryPar\relax%`

17 `\LWR@forcenewpage`

18 `\LWR@atbeginverbatim{programlisting}%`

19

20 `\let\lsthk@EveryPar\LWR@origlsthkEveryPar%`

21 `\else%`

Inline, so open a ``:

```

22 \ifbool{LWR@verbtags}{\LWR@htmltag{span class="inlineprogramlisting"}}{}%
23 \fi%
24 }

```

`\lst@DeInit` Done at the end of a listing.

```

25 \renewcommand*{\lst@DeInit}{%
26 \lst@ifdisplaystyle%

```

Creating a display.

Disable line numbers, produce the `</pre>`, then reenables line numbers:

```

27 \let\lsthk@EveryPar\relax%
28
29 \LWR@afterendverbatim%
30 \let\lsthk@EveryPar\LWR@origlsthkEveryPar%
31 \else%

```

Inline, so create the closing ``:

```

32 \ifbool{LWR@verbtags}{\noindent\LWR@htmltag{/span}}{}%
33 \fi%

```

Final listings deinit:

```

34 \LWR@origlst@DeInit%
35 }

```

`\lst@MakeCaption` `{\t/b}`

This is called BOTH at the top and at the bottom of each listing.

Patched for lwarp.

```

36 \def\lst@MakeCaption#1{%
37 \LWR@traceinfo{MAKING CAPTION at #1}%
38 \lst@ifdisplaystyle
39 \LWR@traceinfo{making a listings display caption}%
40 \ifx #1t%
41 \ifx\lst@caption\@empty\expandafter\lst@HRefStepCounter \else
42 \expandafter\refstepcounter
43 \fi {lstlisting}%
44 \LWR@traceinfo{About to assign label: !\lst@label!}%
45 % \ifx\lst@label\@empty\else
46 % \label{\lst@label}\fi
47 \LWR@traceinfo{Finished assigning the label.}%
48 \let\lst@arg\lst@intname \lst@ReplaceIn\lst@arg\lst@filenamerpl
49 \global\let\lst@name\lst@arg \global\let\lstname\lst@name
50 \lst@ifnolol\else
51 \ifx\lst@caption\@empty
52 \ifx\lst@caption\@empty
53 \ifx\lst@intname\@empty \else \def\lst@temp{ }%
54 \ifx\lst@intname\lst@temp \else

```

This code places a contents entry for a non-float. This would have to be modified for lwarp:

```
55 \LWR@traceinfo{addcontents lst@name: -\lst@name-}%
56 %                \addcontentsline{lol}{lstlisting}{\lst@name}
57                \fi\fi
58                \fi
59                \else
```

This would have to be modified for lwarp:

```
60 \LWR@traceinfo{addcontents lst@@caption: -\lst@@caption-}%
61                \addcontentsline{lol}{lstlisting}%
62 {\protect\numberline{\thelstlisting}}%
63 {\protect\ignorespaces \lst@@caption \protect\relax}}%
64                \fi
65                \fi
66                \fi
67                \ifx\lst@caption\@empty\else
68 \LWR@traceinfo{lst@caption not empty-}%
69                \lst@ifsubstring #1\lst@captionpos
70                {\begingroup
71 \LWR@traceinfo{at the selected position}}%
```

These space and box commands are not needed for HTML output:

```
72 %                \let\@vskip\vskip
73 %                \def\vskip{\afterassignment\lst@vskip \@tempkipa}%
74 %                \def\lst@vskip{\nobreak\@vskip\@tempkipa\nobreak}%
75 %                \par\@parboxrestore\normalsize\normalfont % \noindent (AS)
76 %                \ifx #1t\allowbreak \fi
77                \ifx\lst@title\@empty
```

New lwarp code to create a caption:

```
78                \lst@makecaption\fnum@lstlisting{\ignorespaces \lst@caption}
79                \else
```

New lwarp code to create a title:

```
80 %                \lst@maketitle\lst@title % (AS)
81 \LWR@traceinfo{Making title: \lst@title}%
82 \begin{BlockClass}{lstlistingtitle}% lwarp
83 \lst@maketitle\lst@title% lwarp
84 \end{BlockClass}% lwarp
85                \fi
86 \LWR@traceinfo{About to assign label: !\lst@label!}%
87                \ifx\lst@label\@empty\else
88 \leavevmode% gets rid of bad space factor error
89 \GetTitleStringExpand{\lst@caption}%
90 \edef\LWR@lntemp{\GetTitleStringResult}%
91 \edef\@currentlabelname{\detokenize\expandafter{\LWR@lntemp}}%
92 \label{\lst@label}\fi
93 \LWR@traceinfo{Finished assigning the label.}%
```

Not needed for lwarp:

```

94 %           \ifx #1b\allowbreak \fi
95           \endgroup}{}%
96   \fi
97 \LWR@traceinfo{end of making a listings display caption}%
98   \else
99 \LWR@traceinfo{INLINE}%
100  \fi
101 \LWR@traceinfo{DONE WITH CAPTION at #1}%
102 }
```

Patched to keep left line numbers outside of the left margin, and place right line numbers in a field `\VerbatimHTMLWidth` wide.

```

103 \lst@Key{numbers}{none}{%
104   \let\lst@PlaceNumber\@empty
105   \lstKV@SwitchCases{#1}%
106   {none&\\%
107     left&\def\lst@PlaceNumber{%
108 % \llap{
109 \LWR@orignormalfont%
110 \lst@numberstyle{\thelstnumber}\kern\lst@numbersep%
111 % }
112 }
113 \\%
114   right&\def\lst@PlaceNumber{\rlap{\LWR@orignormalfont
115     \kern\VerbatimHTMLWidth \kern\lst@numbersep
116     \lst@numberstyle{\thelstnumber}}}%
117   }{\PackageError{Listings}{Numbers #1 unknown}\@ehc}}

118 \end{warpHTML}
```

File 84 `lwarp-longtable.sty`

§ 160 Package **longtable**

Pkg `longtable` `longtable` is emulated during HTML output, and the `longtable` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{longtable}`



Longtable `\endhead`, `\endfoot`, and `\endlastfoot` rows are not used for HTML, and these rows should be disabled. Use

```
\warpprintonly{row contents}
```

instead of

```
\begin{warpprint} ... \end{warpprint}
```

Doing so helps avoid “Misplaced \noalign.” when using `\begin{warpprint}`.

Keep the `\endfirsthead` row, which is still relevant to HTML output.

⚠ `\kill` is ignored, place a `\kill` line inside

```
\begin{warpprint} ... \end{warpprint}
```

or place it inside `\warpingprintonly`.

See:

<http://tex.stackexchange.com/questions/43006/why-is-input-not-expandable>

Env `longtable` * [*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{LTchunksiz}
27 \def\endhead{\LWR@tabularendofline}% throws away options //[dim] and /**
28 \def\endfirsthead{\LWR@tabularendofline}
29 \def\endfoot{\LWR@tabularendofline}
30 \def\endlastfoot{\LWR@tabularendofline}
31 \newcommand\tabularnewline{\LWR@tabularendofline}
32 \newcommand{\setlongtables}{}% Obsolete command, does nothing.
33 \newlength{\LTleft}
34 \newlength{\LTright}
35 \newlength{\LTpre}
36 \newlength{\LTpost}
37 \newlength{\LTcapwidth}

38 \renewcommand*{\kill}{\LWR@tabularendofline}

```

File 85 **lwarp-lscape.sty**

§ 161 Package **lscape**

Pkg `lscape` `lscape` is nullified.

for HTML output: Discard all options for `lwarp-lscape`.

```

1 \LWR@ProvidesPackageDrop{lscape}

2 \newenvironment*{landscape}{}{}

```

File 86 **lwarp-ltcaption.sty**

§ 162 Package **ltcaption**

Pkg `ltcaption` `ltcaption` is emulated during HTML output, and the `ltcaption` package is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{ltcaption}`

`\LTcaptype` is already defined by `lwarp`.

`longtable*` is already defined by `lwarp-longtable`.

```

2 \newlength{\LTcapskip}
3 \newlength{\LTcapleft}
4 \newlength{\LTcapright}

```

```
5 \newcommand*{\LTcapmarginfalse}{}

```

File 87 **lwarp-marginfit.sty**

§ 163 Package **marginfit**

Pkg marginfit Emulated.

for HTML output: Discard all options for lwarp-marginfit:

```
1 \LWR@ProvidesPackageDrop{marginfit}

```

File 88 **lwarp-marginfix.sty**

§ 164 Package **marginfix**

Pkg marginfix Emulated.

for HTML output: Discard all options for lwarp-marginfix:

```
1 \LWR@ProvidesPackageDrop{marginfix}

2 \newcommand*{\marginskip}[1]{}
3 \newcommand*{\clearmargin}{}
4 \newcommand*{\softclearmargin}{}
5 \newcommand*{\extendmargin}[1]{}
6 \newcommand*{\mparshift}[1]{}
7 \newdimen\marginheightadjustment
8 \newdimen\marginposadjustment
9 \newcommand*{\blockmargin}[1][1]{}
10 \newcommand*{\unblockmargin}[1][1]{}
11 \newcommand*{\marginphantom}[2][1]{}

```

File 89 **lwarp-marginnote.sty**

§ 165 Package **marginnote**

Pkg marginnote Emulated.

for HTML output: Discard all options for lwarp-marginnote:

```
1 \LWR@ProvidesPackageDrop{marginnote}

```

```

2 \NewDocumentCommand{\marginnote}{o +m o}{\marginpar{#2}}
3 \newcommand*{\marginnoteleftadjust}{}
4 \newcommand*{\marginnoterightadjust}{}
5 \newcommand*{\marginnotetextwidth}{}
6 \let\marginnotetextwidth\textwidth
7 \newcommand*{\marginnotevadjust}{}
8 \newcommand*{\marginfont}{}
9 \newcommand*{\raggedleftmarginnote}{}
10 \newcommand*{\raggedrightmarginnote}{}

```

File 90 **lwarp-mcaption.sty**

§ 166 Package **mcaption**

Pkg mcaption mcaption is nullified.

for HTML output: Discard all options for lwarp-mcaption:

```

1 \LWR@ProvidesPackageDrop{mcaption}

2 \newenvironment{margincap}{}{}
3 \newcommand*{\margincapalign}{}
4 \newlength{\margincapsep}

```

File 91 **lwarp-mdframed.sty**

§ 167 Package **mdframed**

Pkg mdframed mdframed is loaded with options forced to framemethod=none.

for HTML output:

```

1 \RequirePackage{xcolor}% for \convertcolorspec
2 \LWR@ProvidesPackageDrop{mdframed}

```

support Most basic functionality is supported, including frame background colors and single-border colors and thickness, title and subtitle background colors and borders and thickness, border radius, and shadow. CSS classes are created for mdframed environments and frame titles.

 **loading** When used, lwarp loads mdframed in HTML with framemethod=none.

font For title font, use

```
frametitlefont=\textbf,
```

instead of

```
frametitlefont=\bfseries,
```

where `\textbf` must appear just before the comma and will receive the following text as its argument (since the text happens to be between braces in the `mdframed` source). Since `lwarp` does not support `\bfseries` and friends, only one font selection may be made at a time.

theoremtitlefont `theoremtitlefont` is not supported, since the following text is not in braces in the `mdframed` source.

footnotes Footnotes are currently placed at the bottom of the HTML page.

ignored options `userdefinedwidth` and `align` are currently ignored.

CSS classes Environments created or encapsulated by `mdframed` are enclosed in a `<div>` of class `md<environmentname>`, or `mdframed` otherwise.

Frame titles are placed into a `` of class `mdframedtitle`. Subtitles are in a `` of class `mdframedsubtitle`, and likewise for subsubtitles.

Pre-existing hooks are used to patch extra functions before and after the frames.

`amsthm` must be loaded before `mdframed`

```
3 \LWR@origRequirePackage{amsthm}
```

Do not require `Tikz` or `pstricks`:

```
4 \LWR@origRequirePackage[framemethod=none]{mdframed}
```

To handle CSS and paragraphs, patch code at start and end of environment and contents. `\LWR@origraggedright` helps avoid hyphenation.

```
5 \mdfsetup{
6 startcode={\LWR@mdframedstart\LWR@origraggedright},
7 endcode={\LWR@mdframedend},
8 startinnercode={\LWR@startpars\LWR@origraggedright},
9 endinnercode={\LWR@stoppars},
10 }
```

`\LWR@mdfprintcolor` `{<mdfcolorkey>}`

Given the `mdframed` key, print the color.

```
11 \newcommand*{\LWR@mdfprintcolor}[1]{%
12 \convertcolorspec{named}{\csuse{mdf@#1}}{HTML}\LWR@tempcolor%
13 \#\LWR@tempcolor}
```

14 }

`\LWR@mdfprintlength` `{\mdflengthkey}`

Given the mdframed key, print the length.

```
15 \newcommand*\LWR@mdfprintlength}[1]{%
16 \rndprintlength{\csuse{mdf@#1@length}}
17 }
```

`\LWR@mdframedstart` Actions before an mdframe starts.

Encapsulate a frame inside a <div> of the desired class.

```
18 \newcommand*\LWR@mdframedstart}{%
```

Turn off paragraph handling during the generation of the encapsulating tags:

```
19 \LWR@stoppars%
```

Below, print HTML pt units:

```
20 \uselengthunit{PT}%
```

Open a <div> and with custom class and custom style:

```
21 \LWR@htmltagc{div class="\LWR@mdthisenv" \LWR@orignewline
22 style=" \LWR@orignewline
```

Convert and print the background color:

```
23 background: \LWR@mdfprintcolor{backgroundcolor} ; \LWR@orignewline
```

Convert and print the border color and width:

```
24 border: \LWR@mdfprintlength{linewidth} solid
25 \LWR@mdfprintcolor{linecolor} ; \LWR@orignewline
```

Convert and print the border radius:

```
26 border-radius: \LWR@mdfprintlength{roundcorner} ; \LWR@orignewline
```

Convert and print the shadow:

```
27 \ifbool{mdf@shadow}{%
28   box-shadow:
29     \LWR@mdfprintlength{shadowsize}
30     \LWR@mdfprintlength{shadowsize}
31     \LWR@mdfprintlength{shadowsize}
32     \LWR@mdfprintcolor{shadowcolor} ;
33 }
34 {box-shadow: none ;}
35 \LWR@orignewline
```

```
36 "}
```

```
37 % \LWR@htmldivclass{\LWR@mdthisenv}
```

mdframed environment may not work with the HTML versions of the following, so restore them to their originals while inside mdframed:

```
38 \LetLtxMacro{\hspace}{\LWR@origospace}%
39 \LetLtxMacro\rule\LWR@origrule%
40 \LetLtxMacro\makebox\LWR@origmakebox%
41 }
```

`\LWR@mdframedend` Actions after an mdframe ends.

After closing the `<div>`, globally restore to the default environment type:

```
42 \newcommand*{\LWR@mdframedend}{
```

Close the custom `<div>`:

```
43 \LWR@htmldivclassend{\LWR@mdthisenv}
```

Reset future custom class to the default:

```
44 \gdef\LWR@mdthisenv{mdframed}
```

Resume paragraph handling:

```
45 \LWR@startpars%
46 }
```

`\mdfframedtitleenv` `{\title}`

Encapsulation of the original which places the title inside a `` of class `mdframedtitle`:

```
47 \LetLtxMacro\LWR@origmdfframedtitleenv\mdfframedtitleenv
48
49 \newlength{\LWR@titleroundcorner}
50
51 \renewrobustcmd\mdfframedtitleenv[1]{%
52 \LWR@origmdfframedtitleenv{%
```

Below, print HTML pt lengths:

```
53 \uselengthunit{PT}%
```

Open a `` with a custom class and custom style:

```
54 \LWR@htmltagc{span class="mdframedtitle" \LWR@orignewline
55 style=" \LWR@orignewline
```

Convert and print the title background color:

```
56 background:
57 \LWR@mdfprintcolor{frametitlebackgroundcolor}
58 ; \LWR@orignewline
```

Convert and print the title rule:

```
59 \ifbool{mdf@frametitlerule}{%
60   border-bottom:
```

```

61 \LWR@mdfprintlength{frametitlerulewidth}
62 solid
63 \LWR@mdfprintcolor{frametitlerulecolor}
64 ; \LWR@orignewline
65 }{}%

```

The title's top border radius is adjusted for the line width:

```

66 border-radius:
67 \setlength{\LWR@titleroundcorner}
68 {\maxof{\mdf@roundcorner@length-\mdf@linewidth@length}{Opt}}
69 \rndprintlength{\LWR@titleroundcorner}
70 \rndprintlength{\LWR@titleroundcorner}
71 Opt Opt
72 \LWR@orignewline

```

Finish the custom style and the opening span tag:

```

73 " \LWR@orignewline
74 }% span

```

Restrict paragraph tags inside a span:

```

75 \begin{LWR@nestspan}%

```

Print the title inside the span:

```

76 #1%

```

Close the span and unnest the paragraph tag restriction:

```

77 \LWR@htmltagc{/span}%
78 \end{LWR@nestspan}%
79 }
80 }

```

```

\LWR@mdfsubtitlecommon {\sub -or- subsub} [⟨options⟩] {⟨title⟩}

```

Common code for `\LWR@mdfsubtitle` and `\LWR@mdfsubsubtitle`.

Encapsulate the subtitle inside a `` of class `mdframedsubtitle`:

```

81 \NewDocumentCommand{\LWR@mdfsubtitlecommon}{m o m}
82 {% the following empty line is required
83

```

Special handling for `mdframed`: Subtitles have `\pars` around them, so temporarily disable them here.

```

84 \let\par\LWR@origpar%

```

Open a `` with a custom class and custom style:

```

85 \LWR@htmltagc{span class="mdframed#1title"
86 style=" \LWR@orignewline

```

Convert and print the background color:

```
87 background:
88 \LWR@mdfprintcolor{#1titlebackgroundcolor}
89 ; \LWR@orignewline
```

Convert and print the above line:

```
90 \ifbool{mdf@#1titleaboveline}{%
91   border-top:
92   \LWR@mdfprintlength{#1titleabovelinewidth}
93   solid
94   \LWR@mdfprintcolor{#1titleabovelinecolor}
95   ; \LWR@orignewline
96 }{}%
```

Convert and print the below line:

```
97 \ifbool{mdf@#1titlebelowline}{%
98   border-bottom:
99   \LWR@mdfprintlength{#1titlebelowlinewidth}
100   solid
101   \LWR@mdfprintcolor{#1titlebelowlinecolor}
102   ; \LWR@orignewline
103 }{}%
```

Finish the custom style and the opening span tag:

```
104 "% span
```

Restrict paragraph tags inside a span:

```
105 \begin{LWR@nestspan}%
```

Perform the original subtitle action:

```
106 \IfNoValueTF{#2}
107 {\csuse{LWR@origmdf#1title}{#3}}%
108 {\csuse{LWR@origmdf#1title}[#2]{#3}}%
```

Close the span and unnest the paragraph tag restriction:

```
109 \LWR@htmltagc{/span}% the following empty line is required
110 \end{LWR@nestspan}% must follow the /span or an extra <p> appears
111
112 }
```

`\LWR@mdfsubtitle` [*<options>*] {*<title>*}

```
113 \let\LWR@origmdfsubtitle\mdfsubtitle
114
115 \newcommand*{\LWR@mdfsubtitle}{%
116 \LWR@mdfsubtitlecommon{sub}%
117 }
118 \let\mdfsubtitle\LWR@mdfsubtitle
```

```

\LWR@mdfsubsubtitle  [⟨options⟩] {⟨title⟩}
119 \let\LWR@origmdfsubsubtitle\mdfsubsubtitle
120
121 \newcommand*{\LWR@mdfsubsubtitle}{%
122 \LWR@mdfsubsubtitlecommon{subsub}%
123 }
124 \let\mdfsubsubtitle\LWR@mdfsubsubtitle

\LWR@mdthisenv  Stores the environment of the frame about to be created:
125 \newcommand*{\LWR@mdthisenv}{mdframed}

\newmdenv  [⟨options⟩] {⟨env-name⟩}
Modified from the original to remember the environment.
126 \renewrobustcmd*\newmdenv [2] [] {%
127 \newenvironment{#2}%
128 {%
129 \mdfsetup{#1}%
130 \renewcommand*{\LWR@mdthisenv}{md#2}%
131 \begin{mdframed}%
132 }
133 {\end{mdframed}}%
134 }

\surroundwithmdframed  [⟨options⟩] {⟨environment⟩}
Modified from the original to remember the environment.
135 \renewrobustcmd*{\surroundwithmdframed} [2] [] {%
136 \BeforeBeginEnvironment{#2}{%
137 \renewcommand*{\LWR@mdthisenv}{md#2}%
138 \begin{mdframed}[#1]}%
139 \AfterEndEnvironment{#2}{\end{mdframed}}%
140 }

\mdtheorem  [⟨mdframed-options⟩] envname [⟨numberedlike⟩] {⟨caption⟩} [⟨within⟩]
Modified from the original to remember the environment.
141 \DeclareDocumentCommand{\mdtheorem}{ O{} m o m o }%
142 {\ifcsdef{#2}%
143   {\mdf@PackageWarning{Environment #2 already exists\MessageBreak}}%
144   {%
145     \IfNoValueTF {#3}%
146     {%#3 not given -- number relationship
147       \IfNoValueTF {#5}%
148       {%#3+#5 not given

```

```

149     \@definecounter{#2}%
150     \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
151     \newenvironment{#2}[1] [] {%
152         \refstepcounter{#2}%
153         \ifstrempy{##1}%
154             {\let\@temptitle\relax}%
155             {%
156                 \def\@temptitle{\mdf@theoremseparator%
157                     \mdf@theoremspace%
158                     \mdf@theoremtitlefont%
159                     ##1}%
160                 \mdf@thm@caption{#2}{#4}{\csname the#2\endcsname}{##1}}%
161             }%
162     \begin{mdframed}[#1,frametitle={\strut#4 \csname the#2\endcsname%
163         \@temptitle}]]%
164     {\end{mdframed}}%
165     \newenvironment{#2*}[1] [] {%
166         \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
167         \begin{mdframed}[#1,frametitle={\strut#4 \@temptitle}]]%
168         {\end{mdframed}}%
169     }%
170     {%#5 given -- reset counter
171     \@definecounter{#2}\@newctr{#2}[#5]%
172     \expandafter\xdef\csname the#2\endcsname{\@thmcounter{#2}}%
173     \expandafter\xdef\csname the#2\endcsname{%
174         \expandafter\noexpand\csname the#5\endcsname \@thmcountersep%
175         \@thmcounter{#2}}%
176     \newenvironment{#2}[1] [] {%
177         \refstepcounter{#2}%
178         \ifstrempy{##1}%
179             {\let\@temptitle\relax}%
180             {%
181                 \def\@temptitle{\mdf@theoremseparator%
182                     \mdf@theoremspace%
183                     \mdf@theoremtitlefont%
184                     ##1}%
185                 \mdf@thm@caption{#2}{#4}{\csname the#2\endcsname}{##1}}%
186             }%
187     \begin{mdframed}[#1,frametitle={\strut#4 \csname the#2\endcsname%
188         \@temptitle}]]%
189     {\end{mdframed}}%
190     \newenvironment{#2*}[1] [] {%
191         \ifstrempy{##1}%
192             {\let\@temptitle\relax}%
193             {%
194                 \def\@temptitle{\mdf@theoremseparator%
195                     \mdf@theoremspace%
196                     \mdf@theoremtitlefont%
197                     ##1}%
198                 \mdf@thm@caption{#2}{#4}{\csname the#2\endcsname}{##1}}%

```

```

199         }%
200         \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
201         {\end{mdframed}}%
202     }%
203 }%
204 {%#3 given -- number relationship
205     \global\@namedef{the#2}{\@nameuse{the#3}}%
206     \newenvironment{#2}[1] []{%
207         \refstepcounter{#3}%
208         \ifstrempy{##1}%
209             {\let\@temptitle\relax}%
210             {%
211                 \def\@temptitle{\mdf@theoremseparator%
212                     \mdf@theoremspace%
213                     \mdf@theoremtitlefont%
214                     ##1}%
215                 \mdf@thm@caption{#2}{#4}{\csname the#2\endcsname}{##1}}%
216             }
217         \begin{mdframed}[#1,frametitle={\strut#4\ \csname the#2\endcsname%
218             \@temptitle}]]%
219             {\end{mdframed}}%
220         \newenvironment{#2*}[1] []{%
221             \ifstrempy{##1}{\let\@temptitle\relax}{\def\@temptitle{: \ ##1}}%
222             \begin{mdframed}[#1,frametitle={\strut#4\@temptitle}]]%
223             {\end{mdframed}}%
224         }%
225     \BeforeBeginEnvironment{#2}{\renewcommand*\LWR@mdthisenv}{md#2}}% new
226     \BeforeBeginEnvironment{#2*}{\renewcommand*\LWR@mdthisenv}{md#2}}% new
227 }%
228 }

```

`\newmdtheoremenv` [*(mdframed-options)*] envname [*(numberedlike)*] [*(caption)*] [*(within)*]

Modified from the original to remember the environment.

```

229 \DeclareDocumentCommand\newmdtheoremenv{0}{ m o m o }{%
230     \ifboolexpr{ test {\IfNoValueTF {#3}} and test {\IfNoValueTF {#5}} }%
231         {\newtheorem{#2}{#4}}%
232         {%
233             \IfValueT{#3}{\newtheorem{#2}[#3]{#4}}%
234             \IfValueT{#5}{\newtheorem{#2}{#4}[#5]}%
235         }%
236     \BeforeBeginEnvironment{#2}{%
237         \renewcommand*\LWR@mdthisenv}{md#2}}%
238     \begin{mdframed}[#1]}%
239     \AfterEndEnvironment{#2}{%
240     \end{mdframed}}%
241 }

```

 File 92 **lwarp-metalogo.sty**

 § 168 Package **metalogo**

Pkg metalogo metalogo is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{metalogo}

2 \newcommand\setlogokern[2]{}
3 \newcommand\setlogodrop[2][XeTeX]{}
4 \newcommand\setLaTeXa[1]{}
5 \newcommand\setLaTeXee[1]{}
6 \newcommand\seteverylogo[1]{}
7 \newcommand\everylogo[1]{}

```

 File 93 **lwarp-microtype.sty**

 § 169 Package **microtype**

Pkg microtype microtype is pre-loaded by lwarp. All user options and macros are ignored and disabled.

for HTML output: Discard all options for lwarp-microtype:

```

1 \LWR@ProvidesPackageDrop{microtype}

2 \DeclareDocumentCommand{\DeclareMicrotypeSet}{o m m}{}
3 \DeclareDocumentCommand{\UseMicrotypeSet}{o m}{}
4 \DeclareDocumentCommand{\DeclareMicrotypeSetDefault}{o m}{}
5 \DeclareDocumentCommand{\SetProtrusion}{o m m}{}
6 \DeclareDocumentCommand{\SetExpansion}{o m m}{}
7 \DeclareDocumentCommand{\SetTracking}{o m m}{}
8 \DeclareDocumentCommand{\SetExtraKerning}{o m m}{}
9 \DeclareDocumentCommand{\SetExtraSpacing}{o m m}{}
10 \DeclareDocumentCommand{\DisableLigatures}{o m}{}
11 \DeclareDocumentCommand{\DeclareCharacterInheritance}{o m m}{}
12 \DeclareDocumentCommand{\DeclareMicrotypeVariants}{m}{}
13 \DeclareDocumentCommand{\DeclareMicrotypeAlias}{m m}{}
14 \DeclareDocumentCommand{\LoadMicrotypeFile}{m}{}
15 \DeclareDocumentCommand{\DeclareMicrotypeBabelHook}{m m}{}
16 \DeclareDocumentCommand{\microtypesetup}{m}{}
17 \DeclareDocumentCommand{\microtypecontext}{m}{}

```

```

18 \DeclareDocumentCommand{\textmicrotypecontext}{m m}{#2}
19 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
20 \DeclareDocumentCommand{\lsstyle}{}{-}
21 \DeclareDocumentCommand{\textls}{o +m}{-}
22 \DeclareDocumentCommand{\lslig}{m}{#1}
23 }
24 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
25 \def\DeclareMicrotypeVariants#1#{\@gobble}
26 \@onlypreamble\DeclareMicrotypeSet
27 \@onlypreamble\UseMicrotypeSet
28 \@onlypreamble\DeclareMicrotypeSetDefault
29 \@onlypreamble\DisableLigatures
30 \@onlypreamble\DeclareMicrotypeVariants
31 \@onlypreamble\DeclareMicrotypeBabelHook

```

File 94 **lwarp-midfloat.sty**

§ 170 Package **midfloat**

Pkg midfloat midfloat is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{midfloat}

2 \newenvironment{strip}[1][{}]{-}
3 \newskip\stripsep

```

File 95 **lwarp-moreverb.sty**

§ 171 Package **moreverb**

Pkg moreverb moreverb is supported with some patches.

for HTML output:

```

1 \begin{warpHTML}

2 \LWR@ProvidesPackagePass{moreverb}

3 \BeforeBeginEnvironment{verbatim}{-%
4 \LWR@forcenewpage
5 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
6 }
7 \AfterEndEnvironment{verbatim}{-%
8 \LWR@afterendverbatim%
9 }

```

```
10
11
12 \LetLtxMacro\LWRMV@orig@verbatiminput\@verbatiminput
13
14 \renewcommand{\@verbatiminput}[2] [] {%
15 \LWR@forcenewpage
16 \LWR@atbeginverbatim{Verbatim}\unskip\LWR@origvspace*{-\baselineskip}%
17 \LWRMV@orig@verbatiminput[#1]{#2}%
18 \LWR@afterendverbatim%
19 }
20
21 \BeforeBeginEnvironment{listing}{%
22 \LWR@forcenewpage
23 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
24 }
25
26 \AfterEndEnvironment{listing}{%
27 % \unskip\LWR@origvspace*{-\baselineskip}%
28 \LWR@afterendverbatim%
29 }
30
31 \BeforeBeginEnvironment{listingcont}{%
32 \LWR@forcenewpage
33 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
34 }
35
36 \AfterEndEnvironment{listingcont}{%
37 % \unskip\LWR@origvspace*{-\baselineskip}%
38 \LWR@afterendverbatim%
39 }
40
41 \renewcommand{\listinglabel}[1]{\InlineClass{listinglabel}{\the#1}}
42
43 \def\thelisting@line{%
44 \@tempcnta=\listing@line%
45 \divide\@tempcnta\listing@step \multiply\@tempcnta\listing@step%
46 \ifnum\listing@line=\@one%
47 \listinglabel\listing@line%
48 \else%
49 \ifnum\@tempcnta=\listing@line%
50 \listinglabel\listing@line%
51 \else%
52 \InlineClass{listinglabel}{}%
53 \fi%
54 \fi}
55
56 \LetLtxMacro\LWRMV@@listinginput\@listinginput
57
58 \renewcommand{\@listinginput}[3] [] {
59 \LWR@forcenewpage
```

```

60 \LWR@atbeginverbatim{programlisting}\unskip\LWR@origvspace*{-\baselineskip}%
61 \LWRMV@listinginput[#1]{#2}{#3}
62 \LWR@afterendverbatim%
63 }
64
65
66 \renewenvironment*{boxedverbatim}
67 {
68 \LWR@forcenewpage
69 \LWR@atbeginverbatim{boxedverbatim}\unskip\LWR@origvspace*{-\baselineskip}%
70 \verbatim%
71 }
72 {
73 \endverbatim%
74 \LWR@afterendverbatim%
75 }
76
77

78 \end{warpHTML}

```

File 96 **lwarp-mparhack.sty**

§ 172 Package **mparhack**

Pkg mparhack Emulated.

for HTML output: Discard all options for lwarp-mparhack:

```
1 \LWR@ProvidesPackageDrop{mparhack}
```

File 97 **lwarp-multicol.sty**

§ 173 Package **multicol**

Pkg multicol multicol is emulated during HTML output, and the multicol package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{multicol}[2015/09/13]

Multicols are converted into a 1–3 column display, browser-supported.

The optional multicols heading is placed inside a <div> of class multicolsheading.

The content is placed inside a <div> of class multicols.

```

2 \begin{warpHTML}

Env multicols * {<numcols> [<heading>]
3 \NewDocumentEnvironment{multicols}{s m o}

HTML <div> class to contain everything:
4 {
5 \LWR@forcenewpage
6 \BlockClass{multicols}

Optional HTML <div> class for the heading:
7 \IfValueT{#3}{\begin{BlockClass}{multicolshheading}#3\end{BlockClass}}

When done with the environment, close the <div>:
8 {\endBlockClass}

Emulated null functions which are not used in HTML:

9 \newcommand*{\columnbreak}{}
10 \newcommand*{\RLmulticolcolumns}{}
11 \newcommand*{\LRmulticolcolumns}{}
12
13 \newlength{\premulticols}
14 \newlength{\postmulticols}
15 \newlength{\multicolsep}
16 \newlength{\multicolbaselineskip}
17 \newlength{\multicoltolerance}
18 \newlength{\multicolpretolerance}
19 \newcommand*{\columnseprulecolor}{\normalcolor}
20 \newcounter{columnbadness}
21 \newcounter{finalcolumnbadness}
22 \newcounter{collectmore}
23 \newcounter{unbalance}
24 \newlength{\multicolovershoot}
25 \newlength{\multicolundershoot}

26 \end{warpHTML}

```

File 98 **lwarp-multirow.sty**

§ 174 Package **multirow**

Pkg **multirow** multirow is emulated during HTML output, and used as-is while inside a lateximage.

In a lateximage, the original print-mode versions are temporarily restored by `\LWR@restoreorigformatting`.

See section 58.17 for the print-mode versions.

for HTML output: `1 \LWR@ProvidesPackagePass{multirow}`

Remember the print-mode version:

```
2 \LetLtxMacro\LWR@origmultirow\multirow
```

`\LWR@multirowborder` Set to left or right to create a thick border for the cell, for use by bigdelim:

```
3 \newcommand{\LWR@multirowborder}{}
```

§ 174.1 **Multirow**

`\LWR@tabledatamultirowtag` [*vpos*] {*numrows*} [*bigstruts*] {*width*} [*fixup*] {*text*}

```
4 \RenewDocumentCommand{\multirow}{0{c} m o m o +m}%
5 {%
6 \LWR@traceinfo{*** multirow #1 #2 #4}
7 \LWR@maybebenewtablerow%
```

Print the start of a new table data cell:

```
8 \LWR@htmltag{td rowspan="#2" %
```

The vertical alignment, if given:

```
9 \IfValueT{#1}{%
10 \ifstrequal{#1}{b}{style="vertical-align:bottom" }{}%
11 \ifstrequal{#1}{t}{style="vertical-align:top" }{}%
12 }%
```

The left/right border, if given:

```
13 \ifdefvoid{\LWR@multirowborder}{}{%
14 style="border-\LWR@multirowborder: 2px dotted black ; %
15 padding-\LWR@multirowborder: 2px" %
16 }
```

A class adds the column spec and the rule:

```
17 class="td%
```

Append this column's spec:

```
18 \StrChar{\LWR@tablecolspec}{\theLWR@tablecolspos}%
```

If this column has a `cmidrule`, add “rule” to the end of the HTML class tag:

```
19 \ifthenelse{\equal{\LWR@getexparray{\LWR@midrules}{\theLWR@tablecolspos}}{Y}}{rule}{-}%
20 "{}
```

While printing the text, redefine `\` to generate a new line

```
21 \begingroup \LetLtxMacro{\}\{\LWR@endoffline} #6 \endgroup
22 \LWR@stoppars%
23 \global\boolfalse{\LWR@intabularmetadata}%
24 \renewcommand{\LWR@multirowborder}{-}%
25 \LWR@traceinfo{*** multirow done}%
26 }%
```

§ 174.2 Combined multicolumn and multirow

 `\multicolumn &`
`\multirow`

lwarp does not support combining `\multicolumn` and `\multirow`. Use `\multicolumnrow` instead. To create a 2 column, 3 row cell:

```
\multicolumnrow{2}{c}{c}{3}[0]{1in}[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. 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>}
```

```
27 \NewDocumentCommand{\multicolumnrow}{m m O{} m O{} m O{} +m}{%
```

Figure out how many extra HTML columns to add for `@` and `!` columns:

```
28 \LWR@tabularhtmlcolumns{\theLWR@tablecolspos}{#1}
```

Create the multicolumn/multirow tag:

```
29 \begingroup\LetLtxMacro{\}\{\LWR@endoffline}%
30 \LWR@domulticolumn[#3][#4]{\theLWR@tabhtmlcoltotal}{#2}{#8}%
31 \endgroup%
```

Move to the next \LaTeX column:

```
32 \addtocounter{\LWR@tablecolspos}{#1}%
33 \addtocounter{\LWR@tablecolspos}{-1}%
```

Skip any trailing @ or ! columns for this cell:

```
34 \booltrue{LWR@skipatbang}%
35 }

36 \appto{\LWR@restoreorigformatting}{%
37 \LetLtxMacro\multirow\LWR@origmultirow%
38 \renewcommand{\multicolumnrow}{\LWR@origmulticolumnrow}%
39 }
```

File 99 **lwarp-nameref.sty**

§ 175 Package **nameref**

Pkg **nameref** **nameref** is emulated by **lwarp**.

for HTML output: Discard all options for **lwarp-nameref**:

```
1 \typeout{Using the lwarp html version of package 'nameref' -- discarding options.}
2 \typeout{   Are not using ProvidesPackage, so that other packages}
3 \typeout{   do not attempt to patch lwarp's version of 'nameref'.}
4 \DeclareOption*{}
5 \ProcessOptions\relax
```

File 100 **lwarp-needspace.sty**

§ 176 Package **needspace**

Pkg **needspace** **needspace** is not used during HTML conversion.

for HTML output: Discard all options for **lwarp-needspace**:

```
1 \LWR@ProvidesPackageDrop{needspace}
2
3 \newcommand*{\needspace}[1]{ }
4 \DeclareDocumentCommand{\Needspace}{s m}{ }
```

File 101 **lwarp-newclude.sty**

§ 177 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 102 **lwarp-newunicodechar.sty**

§ 178 Package **newunicodechar**

Pkg newunicodechar Error if newunicodechar is loaded after lwarp.

Discard all options for lwarp-newunicodechar:

for HTML output: 1 \LWR@ProvidesPackageDrop{newunicodechar}
2 \LWR@loadbefore{newunicodechar}

File 103 **lwarp-nextpage.sty**

§ 179 Package **nextpage**

Pkg nextpage nextpage is nullified.

for HTML output: Discard all options for lwarp-nextpage.

1 \LWR@ProvidesPackageDrop{nextpage}

2 \newcommand{\cleartoevenpage}[1] [] {}
3 \newcommand{\movetoevenpage}[1] [] {}
4 \newcommand{\cleartooddpage}[1] [] {}
5 \newcommand{\movetooddpage}[1] [] {}

File 104 **lwarp-nonumonpart.sty**

§ 180 Package **nonumonpart**

Pkg nonumonpart nonumonpart is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nonumonpart}

File 105 **lwarp-nopageno.sty**

§ 181 Package **nopageno**

Pkg nopageno nopageno is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{nopageno}

File 106 **lwarp-nowindow.sty**

§ 182 Package **nowindow**

Pkg nowindow nowindow is not used during HTML conversion.

Discard all options for lwarp-nowindow:

for HTML output: 1 \LWR@ProvidesPackageDrop{nowindow}

<code>\nowindow</code>	<code>[\langle lines \rangle]</code>
<code>\setnowindow</code>	<code>[\langle lines \rangle]</code>
	2 \newcommand*{\nowindow}[1] [] {}
	3 \newcommand*{\setnowindow}[1] [] {}
<code>\noclub</code>	<code>[\langle lines \rangle]</code>
<code>\setnoclub</code>	<code>[\langle lines \rangle]</code>
	4 \newcommand*{\noclub}[1] [] {}
	5 \newcommand*{\setnoclub}[1] [] {}

File 107 **lwarp-ntheorem.sty**

§ 183 Package **ntheorem**

(Based on original code by WOLFGANG MAY, ANDREAS SCHEDLER.)

Pkg ntheorem ntheorem is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: `<div>` of class `theorembody<theoremstyle>`

Theorem Header: `` of class `theoremheader<style>`

where `<theoremstyle>` is `plain`, `break`, etc.

⚠ **Font control** This conversion is not total. Font control is via CSS, and the custom \TeX font settings are ignored.

⚠ **Equation numbering** `ntheorem` has a bug with equation numbering in $\mathcal{A}\mathcal{M}\mathcal{S}$ environments when the option `thref` is used. `lwarp` does not share this bug, so equations with `\split`, etc, are numbered correctly with `lwarp`'s HTML output, but not with the print output. It is recommended to use `cleveref` instead of `ntheorem`'s `thref` option.

Options `amsthm` or `standard` choose which set of theorems and proofs to initialize.

⚠ **Disabled options** The options `thmmarks` and `amsmath` are disabled, since they heavily modify the underlying math code. Theorem marks are emulated. The AMS-math modifications are not done.

Option `thref` is disabled because `cleveref` functions are used instead. `\thref` is emulated.

Option `hyperref` is disabled because `lwarp` emulated `hyperref`.

for HTML output: Some disabled options:

```

1 \DeclareOption{thref}{}
2
3
4 \newbool{LWR@theoremmarks}
5 \boolfalse{LWR@theoremmarks}
6
7 \DeclareOption{thmmarks}{
8 \booltrue{LWR@theoremmarks}
9 \newif\ifsetendmark\setendmarktrue
10 }
11
12
13 \newbool{LWR@theoremamsthm}
14 \boolfalse{LWR@theoremamsthm}
15
16 \DeclareOption{amsthm}{\booltrue{LWR@theoremamsthm}}
17
18
19 \DeclareOption{amsmath}{}
20 \DeclareOption{hyperref}{}
21
22
23 \LWR@ProvidesPackagePass{ntheorem}

```

Storage for the style being used for new theorems.

```
24 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
25 \gdef\theoremstyle#1{%
26   \ifundefined{th@#1}{\@warning
27     {Unknown theoremstyle ‘#1’. Using ‘plain’}%
28     \theoremstyle{plain}
29     \renewcommand{\LWR@newtheoremstyle}{plain}% new
30   }%
31   {
32     \theoremstyle{#1}
33     \renewcommand{\LWR@newtheoremstyle}{#1}% new
34   }
35 }
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
36
37 \gdef\xnthm#1#2[#3]{%
38   \ifthm@tempif
39     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
40     \expandafter\ifundefined{c@#1}%
41       {\@definecounter{#1}}{}%
42     \@newctr{#1}[#3]%
43     \expandafter\xdef\csname the#1\endcsname{%
44       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
45       {\noexpand\csname the\theoremnumbering\endcsname{#1}}}%
46     \expandafter\gdef\csname mkheader@#1\endcsname
47       {\csname setparms@#1\endcsname
48        \@thm{#1}{#1}{#2}
49       }%
50     \global\@namedef{end#1}{\@endtheorem}
51     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
52   \fi
53 }
54
55 \gdef\@ynthm#1#2{%
56   \ifthm@tempif
57     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
58     \expandafter\ifundefined{c@#1}%
59       {\@definecounter{#1}}{}%
60     \expandafter\xdef\csname the#1\endcsname
61       {\noexpand\csname the\theoremnumbering\endcsname{#1}}%
62     \expandafter\gdef\csname mkheader@#1\endcsname
63       {\csname setparms@#1\endcsname
```

```

64     \@thm{#1}{#1}{#2}
65     }%
66     \global\@namedef{end#1}{\@endtheorem}
67     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% new
68     \fi
69 }
70
71 \gdef\@othm#1[#2]#3{%
72   \ifundefined{c@#2}{\@nocounterr{#2}}%
73   {\ifthm@tempif
74     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
75     \global\@namedef{the#1}{\@nameuse{the#2}}%
76     \expandafter\protected@xdef\csname num@addtheoremline#1\endcsname{%
77       \noexpand\@num@addtheoremline{#1}{#3}}%
78     \expandafter\protected@xdef\csname nonum@addtheoremline#1\endcsname{%
79       \noexpand\@nonum@addtheoremline{#1}{#3}}%
80     \theoremkeyword{#3}%
81     \expandafter\protected@xdef\csname #1Keyword\endcsname
82       {\the\theoremkeyword}%
83     \expandafter\gdef\csname mkheader@#1\endcsname
84       {\csname setparms@#1\endcsname
85         \@thm{#1}{#2}{#3}
86       }%
87     \global\@namedef{end#1}{\@endtheorem}
88     \AtBeginEnvironment{#1}{\edef\LWR@thmstyle{\csuse{LWR@thmstyle#1}}}% new
89     \fi}
90 }

```

Mimics a float by incrementing the float counter and generating an HTML anchor. These are used for list-of-theorem cross-references.

```

91 \newcommand{\LWR@inctheorem}{%
92   \addtocounter{LWR@thisfloat}{1}%
93   \LWR@stoppars%
94   \LWR@htmltag{a id="autofloat-\arabic{LWR@thisfloat}"{}}\LWR@htmltag{/a}%
95   \LWR@startpars%
96 }

```

The following are patched for css.

These were in individual files thp.sty for plain, thmb.sty for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class theorembody<style>.

Each header is encased in an \InlineClass of class theoremheader<style>.

```

97 \gdef\newtheoremstyle#1#2#3{%
98   \expandafter\ifundefined{th@#1}%
99   {\expandafter\gdef\csname th@#1\endcsname{%

```

```

100   \def\@begintheorem####1####2{%
101   \LWR@forcenewpage% new
102   \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
103   \LWR@inctheorem% new
104   #2}%
105   \def\@opargbegintheorem####1####2####3{%
106   \LWR@forcenewpage% new
107   \BlockClass{theorembody#1}%\LWR@thisthmstyle% new
108   \LWR@inctheorem% new
109   #3}%
110 }%
111 }%
112 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}
113 }
114
115 \renewtheoremstyle{plain}%
116   {\item[\hskip\labelsep \theorem@headerfont
117     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
118   {\item[\hskip\labelsep \theorem@headerfont
119     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
120
121 \renewtheoremstyle{break}%
122   {\item[
123 % \rlap{\vbox{\hbox{
124   \hskip\labelsep \theorem@headerfont
125   \InlineClass{theoremheaderbreak}{##1\ ##2\theorem@separator}\newline
126 % }\hbox{\strut}}}]
127   ]}%
128   {\item[
129 % \rlap{\vbox{\hbox{
130   \hskip\labelsep \theorem@headerfont
131   \InlineClass{theoremheaderbreak}{##1\ ##2\ (##3)\theorem@separator}\newline
132 % }\hbox{\strut}}}]
133   ]}
134
135 \renewtheoremstyle{change}%
136   {\item[\hskip\labelsep
137     \theorem@headerfont
138     \InlineClass{theoremheaderchange}{##2\ ##1\theorem@separator}]}%
139   {\item[\hskip\labelsep
140     \theorem@headerfont
141     \InlineClass{theoremheaderchange}{##2\ ##1\ (##3)\theorem@separator}]}
142
143 \renewtheoremstyle{changebreak}%
144   {\item[
145 % \rlap{\vbox{\hbox{
146   \hskip\labelsep \theorem@headerfont
147   \InlineClass{theoremheaderchangebreak}{##2\ ##1\theorem@separator}\newline
148 % }\hbox{\strut}}}]
149   ]}%

```

```

150  {\item[
151%  \rlap{\vbox{\hbox{
152      \hskip\labelsep \theorem@headerfont
153      \InlineClass{theoremheaderchangebreak}{##2\ ##1\ (##3)\theorem@separator}\newline
154%  }\hbox{\strut}}}]
155  ]}
156
157 \renewtheoremstyle{margin}%
158  {\item[\hskip\labelsep\theorem@headerfont
159      \InlineClass{theoremheadermargin}{##2 \quad ##1\theorem@separator}
160  ]}%
161  {\item[\hskip\labelsep\theorem@headerfont
162      \InlineClass{theoremheadermargin}{##2 \quad ##1\ (##3)\theorem@separator}
163  ]}
164
165 \renewtheoremstyle{marginbreak}%
166  {\item[\hskip\labelsep\theorem@headerfont
167      \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\theorem@separator}\newline
168  ]}%
169  {\item[\hskip\labelsep\theorem@headerfont
170      \InlineClass{theoremheadermarginbreak}{##2 \quad ##1\ (##3)\theorem@separator}\newline
171  ]}
172
173 \renewtheoremstyle{nonumberplain}%
174  {\item[\theorem@headerfont\hskip\labelsep
175      \InlineClass{theoremheaderplain}{##1\theorem@separator}]}%
176  {\item[\theorem@headerfont\hskip \labelsep
177      \InlineClass{theoremheaderplain}{##1\ (##3)\theorem@separator}]}
178
179 \renewtheoremstyle{nonumberbreak}%
180  {\item[
181%  \rlap{\vbox{\hbox{
182      \hskip\labelsep \theorem@headerfont
183      \InlineClass{theoremheaderbreak}{##1\theorem@separator}\newline
184%  }\hbox{\strut}}}]
185  ]}%
186  {\item[
187%  \rlap{\vbox{\hbox{
188      \hskip\labelsep \theorem@headerfont
189      \InlineClass{theoremheaderbreak}{##1\ (##3)\theorem@separator}\newline
190%  }\hbox{\strut}}}]
191  ]}
192
193 \renewtheoremstyle{empty}%
194  {\item[]}%
195  {\item[\theorem@headerfont \hskip\labelsep\relax
196      \InlineClass{theoremheaderplain}{##3}]}
197
198 \renewtheoremstyle{emptybreak}%
199  {\item[]}%

```

```

200 {\item[\theorem@headerfont \hskip\labelsep\relax
201   \InlineClass{theoremheaderplain}{##3}] \ \newline}

```

The following manually adjust the css for the standard configuration objects which are not a purely plain style:

```

202 \ifbool{LWR@theoremamsthm}{}{%
203 % upright text via CSS
204   \newtheoremstyle{plainupright}%
205   {\item[\hskip\labelsep \theorem@headerfont
206     \InlineClass{theoremheaderplain}{##1\ ##2\theorem@separator}]}%
207   {\item[\hskip\labelsep \theorem@headerfont
208     \InlineClass{theoremheaderplain}{##1\ ##2\ (##3)\theorem@separator}]}
209
210 % upright text and small caps header via CSS
211   \newtheoremstyle{nonumberplainuprightsc}%
212   {\item[\theorem@headerfont\hskip\labelsep
213     \InlineClass{theoremheadersc}{##1\theorem@separator}]}%
214   {\item[\theorem@headerfont\hskip \labelsep
215     \InlineClass{theoremheadersc}{##1\ (##3)\theorem@separator}]}

```

The following standard configuration is renewed using the new css:

```

216   \theoremstyle{plainupright}
217   \theorembodyfont{\upshape}
218   \theoremsymbol{\HTMLUnicode{25A1}}% UTF-8 white box
219   \renewtheorem{Example}{Example}
220   \renewtheorem{example}{Example}
221   \renewtheorem{Beispiel}{Beispiel}
222   \renewtheorem{beispiel}{Beispiel}
223   \renewtheorem{Bemerkung}{Bemerkung}
224   \renewtheorem{bemerkung}{Bemerkung}
225   \renewtheorem{Anmerkung}{Anmerkung}
226   \renewtheorem{anmerkung}{Anmerkung}
227   \renewtheorem{Remark}{Remark}
228   \renewtheorem{remark}{Remark}
229   \renewtheorem{Definition}{Definition}
230   \renewtheorem{definition}{Definition}
231
232   \theoremstyle{nonumberplainuprightsc}
233   \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
234   \renewtheorem{Proof}{Proof}
235   \renewtheorem{proof}{Proof}
236   \renewtheorem{Beweis}{Beweis}
237   \renewtheorem{beweis}{Beweis}
238   \qedsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
239
240   \theoremsymbol{}
241 }% not amsthm

```

Only if the amsthm option was given:

```

242 \ifbool{LWR@theoremamsthm}{
243
244 \gdef\th@plain{%
245   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
246   \def\@begintheorem##1##2{%
247     \LWR@forcenewpage% new
248       \BlockClass{theorembodyplain}% new
249       \LWR@intheorem% new
250       \item[\hskip\labelsep
251 % \theorem@headerfont
252 \InlineClass{theoremheaderplain}{##1\ ##2.}
253   ]}%
254   \def\@opargbegintheorem##1##2##3{%
255     \LWR@forcenewpage% new
256       \BlockClass{theorembodyplain}% new
257       \LWR@intheorem% new
258       \item[\hskip\labelsep
259 % \theorem@headerfont
260 \InlineClass{theoremheaderplain}{##1\ ##2\ (##3).}
261   ]}}
262
263 \gdef\th@nonumberplain{%
264   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
265   \def\@begintheorem##1##2{%
266     \LWR@forcenewpage% new
267       \BlockClass{theorembodyplain}% new
268       \LWR@intheorem% new
269       \item[\hskip\labelsep
270 % \theorem@headerfont
271 \InlineClass{theoremheaderplain}{##1.}
272   ]}%
273   \def\@opargbegintheorem##1##2##3{%
274     \LWR@forcenewpage% new
275       \BlockClass{theorembodyplain}% new
276       \LWR@intheorem% new
277       \item[\hskip\labelsep
278 % \theorem@headerfont
279 \InlineClass{theoremheaderplain}{##1\ (##3).}
280   ]}}
281
282 \gdef\th@definition{%
283   \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
284   \def\@begintheorem##1##2{%
285     \LWR@forcenewpage% new
286       \BlockClass{theorembodydefinition}% new
287       \LWR@intheorem% new
288       \item[\hskip\labelsep
289 % \theorem@headerfont

```

```

290 \InlineClass{theoremheaderdefinition}{##1\ ##2.}
291   ]}%
292 \def\@opargbegintheorem##1##2##3{%
293 \LWR@forcenewpage% new
294   \BlockClass{theorembodydefinition}% new
295   \LWR@inctheorem% new
296   \item[\hskip\labelsep
297 % \theorem@headerfont
298 \InlineClass{theoremheaderdefinition}{##1\ ##2\ (##3).}
299   ]}}
300
301 \gdef\th@nonumberdefinition{%
302 \def\theorem@headerfont{\normalfont\bfseries}\normalfont%
303 \def\@begintheorem##1##2{%
304 \LWR@forcenewpage% new
305   \BlockClass{theorembodydefinition}% new
306   \LWR@inctheorem% new
307   \item[\hskip\labelsep
308 % \theorem@headerfont
309 \InlineClass{theoremheaderdefinition}{##1.}
310   ]}%
311 \def\@opargbegintheorem##1##2##3{%
312 \LWR@forcenewpage% new
313   \BlockClass{theorembodydefinition}% new
314   \LWR@inctheorem% new
315   \item[\hskip\labelsep
316 % \theorem@headerfont
317 \InlineClass{theoremheaderdefinition}{##1\ (##3).}
318   ]}}
319
320 \gdef\th@remark{%
321 \def\theorem@headerfont{\itshape}\normalfont%
322 \def\@begintheorem##1##2{%
323 \LWR@forcenewpage% new
324   \BlockClass{theorembodyremark}% new
325   \LWR@inctheorem% new
326   \item[\hskip\labelsep
327 % \theorem@headerfont
328 \InlineClass{theoremheaderremark}{##1\ ##2.}
329   ]}%
330 \def\@opargbegintheorem##1##2##3{%
331 \LWR@forcenewpage% new
332   \BlockClass{theorembodyremark}% new
333   \LWR@inctheorem% new
334   \item[\hskip\labelsep
335 % \theorem@headerfont
336 \InlineClass{theoremheaderremark}{##1\ ##2\ (##3).}
337   ]}}
338
339 \gdef\th@nonumberremark{%

```

```

340 \def\theorem@headerfont{\itshape}\normalfont%
341 \def\@begintheorem##1##2{%
342 \LWR@forcenewpage% new
343     \BlockClass{theorembodyremark}% new
344     \LWR@inctheorem% new
345     \item[\hskip\labelsep
346 % \theorem@headerfont
347 \InlineClass{theoremheaderremark}{##1.}
348     ]}%
349 \def\@opargbegintheorem##1##2##3{%
350 \LWR@forcenewpage% new
351     \BlockClass{theorembodyremark}% new
352     \LWR@inctheorem% new
353     \item[\hskip\labelsep
354 % \theorem@headerfont
355 \InlineClass{theoremheaderremark}{##1\ (##3).}
356     ]}}
357
358 \gdef\th@proof{%
359 \def\theorem@headerfont{\normalfont\bfseries}\itshape%
360 \def\@begintheorem##1##2{%
361 \LWR@forcenewpage% new
362     \BlockClass{theorembodyproof}% new
363     \LWR@inctheorem% new
364     \item[\hskip\labelsep
365 % \theorem@headerfont
366 \InlineClass{theoremheaderproof}{##1.}
367     ]}%
368 \def\@opargbegintheorem##1##2##3{%
369 \LWR@forcenewpage% new
370     \BlockClass{theorembodyproof}% new
371     \LWR@inctheorem% new
372     \item[\hskip\labelsep
373 % \theorem@headerfont
374 \InlineClass{theoremheaderproof}{##1\ (##3).}
375     ]}}
376
377
378
379 \newcounter{proof}%
380 \if@thmmarks
381     \newcounter{currproofctr}%
382     \newcounter{endproofctr}%
383 \fi
384
385 \gdef\proofSymbol{\openbox}
386
387 \newcommand{\proofname}{Proof}
388
389 \newenvironment{proof}[1][\proofname]{

```

```

390 \th@proof
391 \def\theorem@headerfont{\itshape}%
392 \normalfont
393 \theoremsymbol{\HTMLUnicode{220E}}% UTF-8 end-of-proof
394 \@thm{proof}{proof}{#1}
395 }%
396 {\@endtheorem}
397
398 }{}% amsthm option

```

Patched for CSS:

```

399 \let\LWR@origendtheorem\@endtheorem
400 \renewcommand{\@endtheorem}{%
401 \ifbool{LWR@theoremmarks}{%
402 \ifsetendmark%
403 \InlineClass{theoremendmark}{\csname\InTheoType Symbol\endcsname}%
404 \setendmarkfalse%
405 \fi%
406 }{}%
407 \LWR@origendtheorem%
408 \ifbool{LWR@theoremmarks}{\global\setendmarktrue}{}%
409 \endBlockClass%
410 }

411 \gdef\NoEndMark{\global\setendmarkfalse}

```

Redefined to reuse the float mechanism to add list-of-theorem links:

```

\thm@thmline {<1: printed type>} {<2: #>} {<3: optional>} {<4: page>}

412 \renewcommand{\thm@@thmline@noname}[4]{%
413 \hypertocfloat{1}{theorem}{thm}{#2 #3}{}%
414 }
415
416 \renewcommand{\thm@@thmline@name}[4]{%
417 \hypertocfloat{1}{theorem}{thm}{#1 #2 #3}{}%
418 }

```

This was redefined by ntheorem when loaded, so it is now redefined for lwarp:

```

419 \def\thm@@thmline{\thm@@thmline@name}

```

Patch for CSS:

```

420 \def\listtheorems#1{
421 \LWR@htmlclass{nav}{lothm}%
422 \begingroup
423 \c@tocdepth=-2%

```

```

424 \def\thm@list{#1}\thm@processlist
425 \endgroup
426 \LWR@html@elementclassend{nav}{lothm}%
427 }

```

Proof QED symbol:

```

428
429 \newcommand{\qed}{\quad\the\qedsymbol}
430
431 \AtBeginDocument{
432 \def\openbox{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
433 \def\blacksquare{\text{\HTMLUnicode{220E}}}% UTF-8 end-of-proof
434 \def\Box{\text{\HTMLUnicode{25A1}}}% UTF-8 white box
435 }

```

`\thref {<label>}`

```

436 \newcommand*{\thref}[1]{\cref{#1}}

```

File 108 **lwarp-pagenote.sty**

§ 184 Package **pagenote**

Pkg pagenote pagenote works as-is, but the page option is disabled.

for HTML output:

```

1 \DeclareOption{page}{}
2 \LWR@ProvidesPackagePass{pagenote}

```

File 109 **lwarp-paralist.sty**

§ 185 Package **paralist**

Pkg paralist paralist is supported with minor changes.

for HTML output:

```

1 \LWR@ProvidesPackagePass{paralist}

2 \AtBeginEnvironment{compactitem}{\LWR@itemizestart}
3 \AtEndEnvironment{compactitem}{\LWR@itemizeend}
4 \AtBeginEnvironment{compactenum}{\LWR@enumeratestart}
5 \AtEndEnvironment{compactenum}{\LWR@enumerateend}
6 \AtBeginEnvironment{compactdesc}{\LWR@descriptionstart}
7 \AtEndEnvironment{compactdesc}{\LWR@descriptionend}
8 \def\paradescriptionlabel#1{\normalfont\textbf{#1}}

```

File 110 **lwarp-parskip.sty**

§ 186 Package **parskip**

Pkg `parskip` `parskip` is ignored.

for HTML output: Discard all options for `lwarp-parskip`.

```
1 \LWR@ProvidesPackageDrop{parskip}
```

File 111 **lwarp-pdfscape.sty**

§ 187 Package **pdfscape**

Pkg `pdfscape` Emulated.

for HTML output: Discard all options for `lwarp-pdfscape`:

```
1 \LWR@ProvidesPackageDrop{pdfscape}
```

File 112 **lwarp-pdfsync.sty**

§ 188 Package **pdfsync**

Pkg `pdfsync` Emulated.

for HTML output: Discard all options for `lwarp-pdfsync`:

```
1 \LWR@ProvidesPackageDrop{pdfsync}
```

```
2 \let\pdfsync\relax
3 \let\pdfsyncstart\relax
4 \let\pdfsyncstop\relax
```

File 113 `lwarp-pfnote.sty`

§ 189 Package **pfnote**

Pkg `pfnote` `pfnote` is emulated.

 **pfnote numbers** While emulating `pfnote`, `lwarp` is not able to reset HTML footnote numbers per page number to match the printed version, as HTML has no concept of page numbers. `lwarp` therefore uses continuous footnote numbering even for `pfnote`.

for HTML output: `1 \LWR@ProvidesPackageDrop{pfnote}`

File 114 `lwarp-placeins.sty`

§ 190 Package **placeins**

Pkg `placeins` `placeins` is not used during HTML conversion.

Discard all options for `lwarp-placeins`:

for HTML output: `1 \LWR@ProvidesPackageDrop{placeins}`

`2 \newcommand*{\FloatBarrier}{}`

File 115 `lwarp-prelim2e.sty`

§ 191 Package **prelim2e**

Pkg `prelim2e` `Emulated`.

for HTML output: Discard all options for `lwarp-prelim2e`:

`1 \LWR@ProvidesPackageDrop{prelim2e}`

`2 \newcommand{\PrelimText}{}`

`3 \newcommand{\PrelimTextStyle}{}`

`4 \newcommand{\PrelimWords}{}`

File 116 `lwarp-quotchap.sty`

§ 192 Package **quotchap**

Pkg quotchap quotchap is emulated.

(Based on original code by KARSTEN TINNEFELD, JAN KLEVER.)

for HTML output:

```

1 \LWR@ProvidesPackageDrop{quotchap}

2 \newcommand{\@quotchap}{}
3 \newlength{\LWR@quotchapwidth}
4
5 \let\@printcites\relax
6
7 \newcommand*{\@iprintcites}{%
```

Place the quotes inside a `<div>` of class `quotchap`, of the maximum selected width:

```

8 \uselengthunit{PT}%
9 \begin{BlockClass}[max-width: \rndprintlength{\LWR@quotchapwidth}]{quotchap}
10 %\begin{minipage}{\LWR@quotchapwidth}
11 \@quotchap
12 %\end{minipage}
13 \end{BlockClass}
```

Deactivate the quote printing:

```

14 \global\let\@printcites\relax
15 }
16
17 \NewEnviron{savequote}[1][\linewidth]{%
```

Remember the width, adjusted for HTML, and make the length assignment global, per:

<https://tex.stackexchange.com/questions/300823/why-is-setlength-ineffective-inside-a-tabular-environment>

```

18 \setlength{\LWR@quotchapwidth}{#1*2}%
19 \global\LWR@quotchapwidth=\LWR@quotchapwidth%
```

Remember the body, and activate the quote printing:

```

20 \global\let\@quotchap\BODY
```

```
21 \global\let\@printcites\@iprintcites%
22 }
```

The quotation author is placed inside a <div> of class qauthor:

```
23 \newcommand{\qauthor}[1]{\begin{BlockClass}{qauthor}{#1}\end{BlockClass}}
```

\qsetcnfont is ignored:

```
24 \newcommand{\qsetcnfont}[1]{}
```

File 117 **lwarp-ragged2e.sty**

§ 193 Package **ragged2e**

Pkg ragged2e ragged2e is not used during HTML conversion.

Discard all options for lwarp-ragged2e:

for HTML output:

```
1 \LWR@ProvidesPackageDrop{ragged2e}

2 \newcommand*\Centering{\centering}
3 \newcommand*\RaggedLeft{\raggedleft}
4 \newcommand*\RaggedRight{\raggedright}
5 \newcommand*\Justifying{}
6 \newlength\CenteringLeftskip
7 \newlength\RaggedLeftLeftskip
8 \newlength\RaggedRightLeftskip
9 \newlength\CenteringRightskip
10 \newlength\RaggedLeftRightskip
11 \newlength\RaggedRightRightskip
12 \newlength\CenteringParfillskip
13 \newlength\RaggedLeftParfillskip
14 \newlength\RaggedRightParfillskip
15 \newlength\JustifyingParfillskip
16 \newlength\CenteringParindent
17 \newlength\RaggedLeftParindent
18 \newlength\RaggedRightParindent
19 \newlength\JustifyingParindent
20 \newenvironment*{Center}{\center}{\endcenter}
21 \newenvironment*{FlushLeft}{\flushleft}{\endflushleft}
22 \newenvironment*{FlushRight}{\flushright}{\endflushright}
23 \newenvironment*{Justify}{\justifying}{\endjustifying}
```

File 118 `lwarp-realscripts.sty`

§ 194 Package **realscripts**

Pkg realscripts realscripts is emulated. See `lwarp.css` for the `` of class `supsubscript`.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{realscripts}

2 \let\LWR@origtextsuperscript\textsuperscript
3 \let\LWR@origtextsubscript\textsubscript
4
5 \RenewDocumentCommand{\textsuperscript}{s}{\LWR@origtextsuperscript}
6 \RenewDocumentCommand{\textsubscript}{s}{\LWR@origtextsubscript}
7
8 \let\realsuperscript\textsuperscript
9 \let\realsubscript\textsubscript
10
11 \let\fakesuperscript\textsuperscript
12 \let\fakesubscript\textsubscript
13
14 \newlength{\subsupersep}
15
16 \newcommand*{\LWR@realscriptsalign}{}
17
18 \newcommand*{\LWR@setrealscriptsalign}[1]{%
19 \renewcommand*{\LWR@realscriptsalign}{}%
20 \ifthenelse{equal{#1}{c}}{\renewcommand{\LWR@realscriptsalign}{text-align:center;}}{}%
21 \ifthenelse{equal{#1}{r}}{\renewcommand{\LWR@realscriptsalign}{text-align:right;}}{}%
22 }
23
24 \DeclareDocumentCommand \textsubsuperscript {s O{1} mm} {%
25 \LWR@setrealscriptsalign{#2}%
26 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
27 \textsuperscript{#4}\textsubscript{#3}%
28 }%
29 }
30
31 \DeclareDocumentCommand \textsupersubscript {s O{1} mm} {%
32 \LWR@setrealscriptsalign{#2}%
33 \InlineClass[\LWR@realscriptsalign]{supsubscript}{%
34 \textsubscript{#4}\textsuperscript{#3}%
35 }%
36 }

```

File 119 **lwarp-reysize.sty**

§ 195 Package **reysize**

Pkg reysize reysize 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`.

`\reysize`, `\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{reysize}

2 \let\LWR@origreysize\reysize
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\reysize\LWR@origreysize%
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@reysize-temp}
21
22 \renewcommand*{\reysize}[1]{%
23 \renewcommand*{\larger}[1][1]{%
24 \renewcommand*{\smaller}[1][1]{%
25 \renewcommand*{\relscale}[1]{%
26
27 \renewcommand*{\textlarger}[2][1]{%
```

```

28 \setcounter{LWR@relsizetemp}{100+(#1*20)}%
29 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textlarger}{#2}%
30 }
31
32 \renewcommand*\textsmaller[2][1]{%
33 \setcounter{LWR@relsizetemp}{100-(#1*20)}%
34 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textsmaller}{#2}%
35 }
36
37 \renewcommand*\textscale[2]{%
38 \setcounter{LWR@relsizetemp}{100*\real{#1}}%
39 \InlineClass[font-size:\arabic{LWR@relsizetemp}\%]{textscale}{#2}%
40 }

```

File 120 **lwarp-romanbar.sty**

§ 196 Package **romanbar**

Pkg romanbar romanbar is patched for use by lwarp.

An inline class with an overline and underline is used.

for HTML output:

```

1 \LWR@ProvidesPackagePass{romanbar}

2 \DeclareRobustCommand{\Roman@bar}[1]{% #1 is in Roman, i.e. MMXII
3 \InlineClass[%
4   text-decoration: overline underline ;
5 ]{romanbar}{#1}%
6 }

```

File 121 **lwarp-romanbarpagenumber.sty**

§ 197 Package **romanbarpagenumber**

Pkg romanbarpagenumber romanbarpagenumber is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{romanbarpagenumber}

```

File 122 **lwarp-rotating.sty**

§ 198 Package **rotating**

Pkg rotating rotating is emulated during HTML output, and the rotating package is ignored.

All rotations are ignored in HTML output.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotating}

2 \LetLtxMacro\sidewaystable\table
3 \let\endsidewaystable\endtable
4
5 \LetLtxMacro\sidewaysfigure\figure
6 \let\endsidewaysfigure\endfigure
7
8 \newenvironment*{sideways}{-}{-}
9 \newenvironment*{turn}[1]{-}{-}
10 \newenvironment*{rotate}[1]{-}{-}
11 \NewDocumentCommand{\turnbox}{m +m}{#2}
12 \let\rotcaption\caption
13 \let\@makerotcaption\@makecaption

```

File 123 **lwarp-rotfloat.sty**

§ 199 Package **rotfloat**

Pkg rotfloat rotfloat is emulated during HTML output, and the rotfloat package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{rotfloat}
2
3 \RequirePackage{float}

```

`\newfloat` $\langle 1: type \rangle \langle 2: placement \rangle \langle 3: ext \rangle [\langle 4: within \rangle]$

Emulates the `\newfloat` command from the float package. Sideways floats are `\let` to the same as regular floats.

“placement” is ignored.

```

4 \RenewDocumentCommand{\newfloat}{m m m o}{%
5 \IfValueTF{#4}

```

```

6 {
7   \DeclareFloatingEnvironment[fileext=#3,within=#4]{#1}
8 }
9 {
10  \DeclareFloatingEnvironment[fileext=#3]{#1}
11  \DeclareFloatingEnvironment[fileext=#3]{sideways#1}
12 }
13 \csletcs{sideways#1}{#1}
14 \csletcs{endsideways#1}{end#1}

```

newfloat package automatically creates the `\listof` command for new floats, but float does not, so remove `\listof` here in case it is manually created later.

```

15 \cslet{listof#1s}\relax
16 \cslet{listof#1es}\relax
17 }

```

File 124 **lwarp-savetrees.sty**

§ 200 Package **savetrees**

Pkg savetrees Emulated.

for HTML output: Discard all options for lwarp-savetrees:

```

1 \LWR@ProvidesPackageDrop{savetrees}

```

File 125 **lwarp-scalefnt.sty**

§ 201 Package **scalefnt**

Pkg scalefnt scalefnt is ignored.

for HTML output: `1 \LWR@ProvidesPackageDrop{scalefnt}`

```

2 \DeclareRobustCommand\scalefont[1]{ }

```

File 126 **lwarp-sectsty.sty**

§ 202 Package **sectsty**

Pkg sectsty sectsty is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{sectsty}

2 \newcommand*{\partfont} [1] {}
3 \newcommand*{\partnumberfont} [1] {}
4 \newcommand*{\parttitlefont} [1] {}
5 \newcommand*{\chapterfont} [1] {}
6 \newcommand*{\chapternumberfont} [1] {}
7 \newcommand*{\chaptertitlefont} [1] {}
8 \newcommand*{\sectionfont} [1] {}
9 \newcommand*{\subsectionfont} [1] {}
10 \newcommand*{\subsubsectionfont} [1] {}
11 \newcommand*{\paragraphfont} [1] {}
12 \newcommand*{\subparagraphfont} [1] {}
13 \newcommand*{\minisecfont} [1] {}
14 \newcommand*{\allsectionsfont}[1] {}
15 \newcommand{\nohang}{}

```

`\sectionrule` is only to be used in `*font` commands, thus it is ignored.

```

16 \newcommand*{\sectionrule}[5]{}
17
18 \def\ulemheading#1#2{}

```

File 127 **lwarp-setspace.sty**

§ 203 Package **setspace**

Pkg `setspace` `setspace` is not used during HTML conversion.

Discard all options for `lwarp-setspace`:

for HTML output:

```

1 \LWR@ProvidesPackageDrop{setspace}
2
3 \newcommand*{\setstretch}[1]{}
4 \newcommand*{\SetSpacespace}[1]{}
5 \newcommand*{\singlespacing}{}
6 \newcommand*{\onehalfspacing}{}
7 \newcommand*{\doublespacing}{}
8
9 \newenvironment*{singlespace}
10 {
11 \LWR@forcenewpage
12 \BlockClass{singlespace}
13 }
14 {\endBlockClass}
15
16 \newenvironment*{singlespace*}

```

```

17 {
18 \LWR@forcenewpage
19 \BlockClass{singlespace}
20 }
21 {\endBlockClass}
22
23 \newenvironment*{spacing}[1]{
24
25 }{
26
27 }
28
29 \newenvironment*{onehalfspace}
30 {
31 \LWR@forcenewpage
32 \BlockClass{onehalfspace}
33 }
34 {\endBlockClass}
35
36 \newenvironment*{doublespace}
37 {
38 \LWR@forcenewpage
39 \BlockClass{doublespace}
40 }
41 {\endBlockClass}

```

File 128 **lwarp-shadow.sty**

§ 204 Package **shadow**

Pkg shadow shadow is emulated.

for HTML output: Discard all options for lwarp-shadow:

```

1 \LWR@ProvidesPackageDrop{shadow}

2 \newdimen\sboxsep
3 \newdimen\sboxrule
4 \newdimen\sdim
5
6 \newcommand{\shabox}[1]{%
7 \InlineClass{shabox}{#1}%
8 }

```

File 129 `lwarp-showidx.sty`

§ 205 Package **showidx**

Pkg `showidx` `showidx` is ignored.

for HTML output: Discard all options for `lwarp-showidx`:

```
1 \LWR@ProvidesPackageDrop{showidx}
```

File 130 `lwarp-showkeys.sty`

§ 206 Package **showkeys**

Pkg `showkeys` `showkeys` is ignored.

for HTML output: Discard all options for `lwarp-showkeys`:

```
1 \LWR@ProvidesPackageDrop{showkeys}
```

```
2 \NewDocumentCommand{\showkeys}{s}{}

```

File 131 `lwarp-sidecap.sty`

§ 207 Package **sidecap**

Pkg `sidecap` `sidecap` is nullified.

for HTML output: Discard all options for `lwarp-sidecap`.

```
1 \LWR@ProvidesPackageDrop{sidecap}
```

See:

<http://tex.stackexchange.com/questions/45401/use-the-s-star-argument-with-newdocumentenvironment>
regarding the creation of starred environments with `xparse`.

```
2 \NewDocumentEnvironment{SCTable}{soo}
3 {\IfValueTF{#3}{\table[#3]}{\table}}
4 {\endtable}
```

```

5
6 \ExplSyntaxOn
7 \cs_new:cpn {SCTable*} {\SCTable*}
8 \cs_new_eq:cN {endSCTable*} \endSCTable
9 \ExplSyntaxOff
10
11
12 \NewDocumentEnvironment{SCfigure}{soo}
13 {\IfValueTF{#3}{\figure[#3]}{\figure}}
14 {\endfigure}
15
16 \ExplSyntaxOn
17 \cs_new:cpn {SCfigure*} {\SCfigure*}
18 \cs_new_eq:cN {endSCfigure*} \endSCfigure
19 \ExplSyntaxOff
20
21
22 \newenvironment*{wide}{}{}

```

File 132 **lwarp-sidenotes.sty**

§ 208 Package **sidenotes**

(Based on original code by ANDY THOMAS, OLIVER SCHEBAUM.)

Pkg sidenotes Patched for lwarp.

for HTML output: Load the original package:

```
1 \LWR@ProvidesPackagePass{sidenotes}
```

The following patch sidenotes for use with lwarp:

Stop paragraph handling while creating the caption:

```

2 \RenewDocumentCommand \sidecaption {s o o m}
3 {
4   \LWR@stoppars
5   \captionsetup{style=sidecaption}
6   \IfBooleanTF{#1}
7   { % starred
8     \IfNoValueOrEmptyTF{#2}
9     {\marginnote{\caption*{#4}}}
10    {\marginnote{\caption*{#4}}[#2]}
11  }
12  { % unstarred
13    \IfNoValueOrEmptyTF{#2}
14    {\def\@sidenotes@sidecaption@tof{#4}}

```

```

15   {\def\@sidenotes@sidecaption@tof{#2}}
16   \IfNoValueOrEmptyTF{#3}
17     {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}}
18     {\marginnote{\caption[\@sidenotes@sidecaption@tof]{#4}}[#3]}
19   }
20   \LWR@startpars
21 }

```

Borrowed from the lwarp version of keyfloat:

```

22 \NewDocumentEnvironment{KFLTsidenotes@marginfloat}{0{-1.2ex} m}
23 {% start
24 \LWR@maybeinthisfloat%
25 \LWR@forcenewpage
26 \LWR@stoppars%
27 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{LWR@thisfloat}"}
28 \LWR@startpars%
29 \captionsetup{type=#2}%
30 }
31 {
32 \LWR@htmldivclassend{div}
33 }
34
35 \RenewDocumentEnvironment{marginfigure}{o}
36   {\begin{KFLTsidenotes@marginfloat}{figure}}
37   {\end{KFLTsidenotes@marginfloat}}
38
39 \RenewDocumentEnvironment{margintable}{o}
40   {\begin{KFLTsidenotes@marginfloat}{table}}
41   {\end{KFLTsidenotes@marginfloat}}

```

The following were changed by sidenotes, and now are reset back to their lwarp-supported originals:

Restoring the definition from the $\LaTeX 2_{\epsilon}$ `article.cls` source:

```

42 \renewenvironment{figure*}
43     {\@dblfloat{figure}}
44     {\end@dblfloat}
45
46 \renewenvironment{table*}
47     {\@dblfloat{table}}
48     {\end@dblfloat}

```

File 133 **lwarp-siunitx.sty**

§ 209 Package **siunitx**

Pkg siunitx siunitx is patched for use by lwarp.



per-mode

Do not use `per-mode=fraction`, which cannot be seen by the final `pdftotext` conversion.

Some units will require that the expression be placed inside math mode.

for HTML output:

```
1 \LWR@ProvidesPackagePass{siunitx}

2 \AtBeginDocument{% in case textcomp was not loaded
3 \DeclareSIUnit\bohr{\textit{a}\textsubscript{0}}
4 \DeclareSIUnit\clight{\textit{c}\textsubscript{0}}
5 \DeclareSIUnit\elementarycharge{\textit{e}}
6 \DeclareSIUnit\electronmass{\textit{m}\textsubscript{e}}
7 \DeclareSIUnit\hartree{\textit{E}\textsubscript{h}}
8 }% AtBeginDocument
9
```

File 134 **lwarp-soul.sty**

§ 210 Package **soul**

(Based on original code by MELCHIOR FRANZ.)

Pkg soul Emulated.

for HTML output:

```
1 \LWR@ProvidesPackageDrop{soul}
2 \RequirePackage{xcolor}% for \convertcolorspec
```

Storage for the colors to use:

```
3 \newcommand*\LWR@soululcolor{}
4
5 \newcommand*\LWR@soulstcolor{}
6
7 % \definecolor{LWR@soulhlcolordefault}{HTML}{F8E800}
8 % \newcommand*\LWR@soulhlcolor{LWR@soulhlcolordefault}
9 \newcommand*\LWR@soulhlcolor{}
```

Basic markup with css:

```
10 \newcommand{\so}[1]{\InlineClass{letterspacing}{#1}}
11 \newcommand{\caps}[1]{\InlineClass{capsspacing}{#1}}
```

Add colors if not empty:

```
12 \newcommand{\LWR@soulcolor}[4]{%
13 \ifcseempty{#2}%
14 {%
15 \InlineClass{#3}{#1}}%
16 {%
17 \convertcolorspec{named}{\csuse{#2}}{HTML}\LWR@tempcolor%
18 \InlineClass[#4: \#\LWR@tempcolor]{#3}{#1}%
19 }%
20 }
21
22 \newcommand{\ul}[1]{%
23 \LWR@soulcolor{#1}{LWR@soululcolor}{uline}{text-decoration-color}%
24 }
25
26 \newcommand{\st}[1]{
27 \LWR@soulcolor{#1}{LWR@soulstcolor}{sout}{text-decoration-color}%
28 }
29
30 \newcommand{\hl}[1]{
31 \LWR@soulcolor{#1}{LWR@soulhlcolor}{highlight}{background-color}%
32 }
```

Nullified:

```
33 \newcommand*{\soulaccent}[1]{}
34 \newcommand*{\soulregister}[2]{}
35 \newcommand{\sloppyword}[1]{#1}
36 \newcommand*{\sodef}[5]{\DeclareRobustCommand*#1[1]{\so{##1}}}
37 \newcommand*{\resetso}{}
38 \newcommand*{\capsdef}[5]{}
39 \newcommand*{\capsreset}{}
40 \newcommand*{\capssave}[1]{}
41 \newcommand*{\capssselect}[1]{}
42 \newcommand*{\setul}[2]{}
43 \newcommand*{\resetul}{}
44 \newcommand*{\setuldepth}[1]{}
45 \newcommand*{\setuloverlap}[1]{}

```

Set colors:

```
46 \newcommand*{\setulcolor}[1]{\renewcommand{\LWR@soululcolor}{#1}}
47 \newcommand*{\setstcolor}[1]{\renewcommand{\LWR@soulstcolor}{#1}}
```

```
48 \newcommand*\sethlcolor}[1]{\renewcommand{\LWR@soulhlcolor}{#1}}
```

Long versions of the user-level macros:

```
49 \let\textso\so
50 \let\textul\ul
51 \let\texthl\hl
52 \let\textcaps\caps
```

File 135 **lwarp-stabular.sty**

§ 211 Package **stabular**

Pkg stabular stabular is emulated.

for HTML output: 1 \LWR@ProvidesPackageDrop{stabular}

Env stabular [*vpos*] {*colspec*}

```
2 \newenvironment{stabular}[2][c]
3 {
4 \renewcommand{\noalign}[1]{-}
5 \begin{tabular}[#1]{#2}
6 }
7 {\end{tabular}}
```

Env stabular {*width*} [*vpos*] {*colspec*}

```
8 \NewDocumentEnvironment{stabular*}{m o m}
9 {
10 \renewcommand{\noalign}[1]{-}
11 \begin{tabular}[#2]{#3}
12 }
13 {\end{tabular}}
```

File 136 **lwarp-subfig.sty**

§ 212 Package **subfig**

(Based on original code by STEVEN DOUGLAS COCHRAN.)

Pkg subfig subfig is supported and patched by lwarp.

 **lof/lotdepth** At present, the package options for lofdepth and lotdepth are not working. These

counters must be set separately after the package has been loaded.

horizontal spacing In the document source, use `\hfill` and `\hspace*` between subfigures to spread them apart horizontally. The use of other forms of whitespace may cause paragraph tags to be generated, resulting in subfigures appearing on the following lines instead of all on a single line.

for HTML output: Accept all options for `lwarp-subfig`:

```
1 \LWR@ProvidesPackagePass{subfig}
```

```
\sf@@@subfloat {<1 type>} [<2 lof entry>] [<3 caption>] {<4 contents>}
```

The outer minipage allows side-by-side subfloats with `\hfill` between.

```
2 \long\def\sf@@@subfloat#1[#2][#3]#4{%
3 \begin{minipage}{\linewidth}% new
4 \LWR@stoppars% new
5   \ifundefined{FBsc@max}{}%
6     {\FB@readaux{\let\FBsuboheight\relax}}%
7   \@tempcnta=\@ne
8   \if@minipage
9     \@tempcnta=\z@
10  \else\ifdim \lastskip=\z@ \else
11    \@tempcnta=\tw@
12  \fi\fi
13  \ifmaincaptiontop
14    \sf@top=\sf@nearskip
15    \sf@bottom=\sf@farskip
16  \else
17    \sf@top=\sf@farskip
18    \sf@bottom=\sf@nearskip
19  \fi
20  \leavevmode
21  \setbox\@tempboxa \hbox{#4}%
22  \@tempdima=\wd\@tempboxa
23  \ifundefined{FBsc@max}{}%
24    {\global\advance\Xhsize-\wd\@tempboxa
25     \dimen@=\ht\@tempboxa
26     \advance\dimen@\dp\@tempboxa
27     \ifdim\dimen@>\FBso@max
28       \global\FBso@max\dimen@
29     \fi}%
30  \vtop\bgroup
31  \vbox\bgroup
32    \ifcase\@tempcnta
33      \@minipagefalse
34    \or
35      \vskip\sf@top
36    \or
```

```

37         \ifdim \lastskip=\z@ \else
38             \@tempskipb\sf@top\relax\@xaddvskip
39         \fi
40     \fi
41     \sf@ifpositiontop{%
42         \ifx \@empty#3\relax \else
43             \sf@subcaption{#1}{#2}{#3}%
44             \vskip\sf@capskip
45             \vskip\sf@captopadj
46         \fi\egroup
47         \hrule widthOpt heightOpt depthOpt
48         \LWR@startpars% new
49 % \box\@tempboxa
50     #4
51         \LWR@stoppars% new
52     }{%
53         \LWR@startpars% new
54         \@ifundefined{FBsc@max}%
55         {
56 % \box\@tempboxa
57     #4
58         }%
59         {\ifx\FBsuboheight\relax
60 %         \box\@tempboxa
61     #4
62         \else
63 %         \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
64     #4
65         \fi}%
66     \LWR@stoppars% new
67     \egroup
68     \ifx \@empty#3\relax \else
69         \vskip\sf@capskip
70         \hrule widthOpt heightOpt depthOpt
71         \sf@subcaption{#1}{#2}{#3}%
72     \fi
73     }%
74     \vskip\sf@bottom
75 \egroup
76 \@ifundefined{FBsc@max}{%
77     {\addtocounter{FRobj}{-1}%
78     \ifnum\c@FRobj=0\else
79         \subfloatrowsep
80     \fi}%
81     \ifmaincaptiontop\else
82     \global\advance\@nameuse{c@\@capttype}\m@ne
83     \fi
84 \end{minipage}% new
85 \LWR@startpars% new
86 \endgroup\ignorespaces%

```

```

87 }%

\sf@subcaption {<1 type>} {<2 lof entry>} {<3 caption>}

88 \long\def\sf@subcaption#1#2#3{%
89 \LWR@stoppars% new
90 \ifx \relax#2\relax \else
91   \bgroup
92     \let\label=\@gobble
93     \let\protect=\string
94     \def\@subcaplabel{%
95       \caption@lstfmt{\@nameuse{p#1}}{\@nameuse{the#1}}}%
96     \sf@updatecaptionlist{#1}{#2}{\the\value{\@capttype}}{\the\value{#1}}%
97   \egroup
98 \fi
99 \bgroup
100 \ifx \relax#3\relax
101   \let\captionlabelsep=\relax
102 \fi
103 %   \setbox0\vbox{%
104 %     \hb@xt@\the\@tempdima{%
105 %
106 % %       \hss
107 % %       \parbox[t]{\the\@tempdima}{%
108 % %         \caption@make
109 % %           {\@nameuse{sub\@capttype name}}%
110 % %           {\@nameuse{thesub\@capttype}}%
111 % %           {#3}
112 % % }%
113 % %       \hss
114 % %   }
115 % % }%
116 %   \@ifundefined{FBsc@max}%
117 %     {\box0}%
118 %     {
119 %       \parbox[t]{\the\@tempdima}{%
120 \LWR@traceinfo{sfsubcap B1}% new
121         \LWR@htmlblocktag{figcaption}% new
122         \caption@make
123           {\@nameuse{sub\@capttype name}}%
124           {\@nameuse{thesub\@capttype}}%
125           {#3}
126         \LWR@htmlblocktag{/figcaption}% new
127 \LWR@traceinfo{sfsubcap B2}% new
128 %     }%
129 %   }%
130 %   {\dimen@ht0%
131 %     \advance\dimen@\dp0%
132 %     \ifdim\dimen@>\FBsc@max
133 %       \global\FBsc@max\dimen@

```

```

134     \fi
135     \FB@readaux{\let\FBsubcheight\relax}%
136     \ifx\FBsubcheight\relax
137         \def\next{
138 % \parbox[t]{\the\@tempdima}
139         }%
140     \else
141         \def\next{
142 % \parbox[t][\FBsubcheight][t]{\the\@tempdima}
143         }%
144     \fi
145     \vbox{%
146 %         \hb@xt@\the\@tempdima{%
147
148 %             \hss
149 %             \next{%
150 \LWR@traceinfo{sfsubcap C1}% new
151             \caption@make
152                 {\@nameuse{sub\@capttype name}}}%
153                 {\@nameuse{thesub\@capttype}}}%
154                 {#3}
155 \LWR@traceinfo{sfsubcap C1}% new
156 % }%
157 %             \hss
158
159 % }
160     }
161     }%
162 \egroup
163 \LWR@startpars% new
164 }

\caption@@@make {<caption label>} {<caption text>}

165 \renewcommand\caption@@@make [2] {%
166 \LWR@startpars% new
167 \sbox\@tempboxa{#1}%
168 \ifdim\wd\@tempboxa=\z@
169 \let\caption@lsep\relax
170 \fi
171 \caption@ifempty{#2}{%
172 \let\caption@lsep\@empty
173 \let\caption@tfmt\@firstofone
174 }%
175 % \setpar{\@par\caption@@par}\caption@@par
176 \renewcommand{\@par}{\LWR@closeparagraph\LWR@orig@@par}% new
177 \caption@applyfont
178 \caption@fmt
179 {\ifcaption@star\else
180 \beginngroup

```

```

181     \captionlabelfont
182     #1%
183     \endgroup
184     \fi}%
185     {\ifcaption@star\else
186     \begingroup
187     \caption@iflf\captionlabelfont
188     \relax\caption@lsep
189     \endgroup
190     \fi}%
191     {\captiontextfont
192     \caption@ifstrut
193     {\vrule\@height\ht\strutbox\@width\z@}%
194     }%
195     \nobreak\hskip\z@skip % enable hyphenation
196     \caption@tfmt{#2}
197     \LWR@ensuredoingapar% new
198     \caption@ifstrut
199     {\ifhmode\@finalstrut\strutbox\fi}%
200     }%
201     \par}}
202 \LWR@stoppars% new
203 }

```

Patches for `\sf@sub@label`:

```

204 \def\subfloat@label{%
205 \LWR@ensuredoingapar% new
206 \@ifnextchar(% %) match left parenthesis
207   {\sf@sub@label}
208   {\sf@sub@label(Sub\@capytype\space
209                 \ifundefined{thechapter}{-}{\@nameuse{thechapter}\space}%
210                 \@nameuse{p@sub\@capytype}%
211                 \@nameuse{thesub\@capytype}.)}}

```

Patches for `\subref`.

The unstarred version uses a `\ref` link whose printed text comes from the `sub@<label>`:

```

212 \renewcommand{\sf@subref}[1]{%
213 \LWR@subnewref{#1}{sub@#1}%
214 }

```

The starred version uses the printed `sub@<label>` which is stored as if it were a page number:

```

215 \renewcommand{\sf@subref}[1]{\LWR@origpageref{sub@#1}}

```

Defining new subfloats. The `l@sub<type>` for each is redefined.

```
216 \LetLtxMacro\LWR@orig@newsfloat\@newsfloat
217
218 \def\@newsfloat[#1]#2{%
219 \LWR@orig@newsfloat[#1]{#2}%
220 \renewcommand{\l@sub#2}[2]{\hypertocfloat{2}{sub#2}{\ext@sub#2}{##1}{##2}}
221 }
```

Pre-defined for figures and tables:

```
222 \renewcommand{\l@subfigure}[2]{\hypertocfloat{2}{subfigure}{lof}{#1}{#2}}
223 \renewcommand{\l@subtable}[2]{\hypertocfloat{2}{subtable}{lot}{#1}{#2}}
224 % \def\subfigure{\subfloat}
225 % \def\subtable{\subfloat}
```

File 137 **lwarp-supertabular.sty**

§ 213 Package **supertabular**

Pkg supertabular supertabular is emulated during HTML output.

for HTML output: `1 \LWR@ProvidesPackageDrop{supertabular}`

 **misplaced alignment alignment tab character &** For `\tablefirsthead`, etc., enclose them as follows:

```
\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 7.7.

```
2 \newcommand{\LWRST@firsthead}{}
3
4 \newcommand{\tablefirsthead}[1]{%
5   \long\gdef\LWRST@firsthead{#1}%
6 }
7
8 \newcommand{\tablehead}[1]{}
9 \newcommand{\tabletail}[1]{}
10
11 \newcommand{\LWRST@lasttail}{}
12
13 \newcommand{\tablelasttail}[1]{%
14   \long\gdef\LWRST@lasttail{#1}%
15 }
```

```

16
17 \newcommand{\tablecaption}[2] [] {%
18   \long\gdef\LWRST@caption{\caption[#1]{#2}}%
19 }
20
21 \let\topcaption\tablecaption
22 \let\bottomcaption\tablecaption
23
24 \global\let\LWRST@caption\relax
25
26 \newcommand*{\shrinkheight}[1] {}
27
28 \NewDocumentEnvironment{supertabular}{s o m}
29 {
30 \LWR@traceinfo{supertabular}
31 \table
32 \LWRST@caption
33 \begin{tabular}{#3}
34 \TabularMacro\ifdefvoid{\LWRST@firsthead}%
35 {\LWR@getmynexttoken}%
36 {\expandafter\LWR@getmynexttoken\LWRST@firsthead}%
37 }
38 {%
39 \ifdefvoid{\LWRST@lasttail}%
40 {}%
41 {%
42 \TabularMacro\ResumeTabular%
43 \LWRST@lasttail%
44 }%
45 \end{tabular}
46 \endtable
47 \LWR@traceinfo{supertabular done}
48 }
49
50 \NewDocumentEnvironment{mpsupertabular}{s o m}
51 {\minipage{\linewidth}\supertabular{#3}}
52 {\endsupertabular\endminipage}

```

File 138 **lwarp-syntonly.sty**

§ 214 Package **syntonly**

Pkg syntonly Emulated.

for HTML output: Discard all options for lwarp-syntonly:

```
1 \LWR@ProvidesPackageDrop{syntonly}
```

```

2 \newif\ifsyntax@
3 \syntax@false
4
5 \newcommand*{\syntaxonly}{\}
6
7 \@onlypreamble\syntaxonly

```

File 139 **lwarp-tables.sty**

§ 215 Package **tables**

Pkg tables tables is emulated. \LWR@hline is used to handle the optional argument when tables is loaded.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{tables}

2 \newdimen\tablinesep
3 \newdimen\arraylinesep
4 \newdimen\extrarulesep

```

File 140 **lwarp-tabularx.sty**

§ 216 Package **tabularx**

Pkg tabularx tabularx is emulated by lwarp.

for HTML output: Discard all options for lwarp-tabularx:

```

1 \LWR@ProvidesPackageDrop{tabularx}

2 \NewDocumentEnvironment{tabularx}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabularx*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}

```

File 141 **lwarp-tabulary.sty**

§ 217 Package **tabulary**

Pkg `tabulary` `tabulary` is emulated by `lwarp`.

for HTML output: Discard all options for `lwarp-tabulary`.

Column types L, C, R, and J are emulated by `lwarp` core code.

```

1 \LWR@ProvidesPackageDrop{tabulary}

2 \NewDocumentEnvironment{tabulary}{m o m}
3 {\tabular{#3}}
4 {\endtabular}
5
6 \NewDocumentEnvironment{tabulary*}{m o m}
7 {\tabular{#3}}
8 {\endtabular}
9
10 \newdimen\tymin
11 \newdimen\tymax
12 \def\tyformat{}
```

File 142 **lwarp-textarea.sty**

§ 218 Package **textarea**

Pkg `textarea` `textarea` is emulated.

for HTML output: `1 \LWR@ProvidesPackageDrop{textarea}`

```

2 \newcommand\StartFromTextArea{}
3 \newcommand\StartFromHeaderArea{}
4 \newcommand*\RestoreTextArea{}
5 \newcommand*\ExpandTextArea[1][*]{}
6 \let\NCC@restorettextarea\@empty
```

File 143 `lwarp-textcomp.sty`

§ 219 Package **textcomp**

Pkg `textcomp` `textcomp` is patched for use by `lwarp`.

Some `textcomp` symbols do not have Unicode equivalents, and thus are not supported.

 **Missing symbols** Many `textcomp` symbols are not supported by many fonts. Try using more complete fonts in the CSS, but expect to see gaps in coverage.

for HTML output: `1 \LWR@ProvidesPackagePass{textcomp}`

The following are restored for print when inside a `lateximage`:

```

2 \let\LWR@origtextdegree\textdegree
3 \let\LWR@origtextcelsius\textcelsius
4 \let\LWR@origtextohm\textohm
5 \let\LWR@origtextmu\textmu
6 \let\LWR@origtextlquill\textlquill
7 \let\LWR@origtextrquill\textrquill
8 \let\LWR@origtextcircledP\textcircledP
9 \let\LWR@origtexttwelveudash\texttwelveudash
10 \let\LWR@origtextthreequartersemdash\textthreequartersemdash
11 \let\LWR@origtextmho\textmho
12 \let\LWR@origtextnaira\textnaira
13 \let\LWR@origtextpeso\textpeso
14 \let\LWR@origtextrecipe\textrecipe
15 \let\LWR@origtextpertenthousand\textpertenthousand
16 \let\LWR@origtextbaht\textbaht
17 \let\LWR@origtextdiscount\textdiscount
18 \let\LWR@origtextservicemark\textservicemark
19 \LetLtxMacro\LWR@origcapitalcedilla\capitalcedilla
20 \LetLtxMacro\LWR@origcapitalogonek\capitalogonek
21 \LetLtxMacro\LWR@origcapitalgrave\capitalgrave
22 \LetLtxMacro\LWR@origcapitalacute\capitalacute
23 \LetLtxMacro\LWR@origcapitalcircumflex\capitalcircumflex
24 \LetLtxMacro\LWR@origcapitaltilde\capitaltilde
25 \LetLtxMacro\LWR@origcapitaldieresis\capitaldieresis
26 \LetLtxMacro\LWR@origcapitalhungarumlaut\capitalhungarumlaut
27 \LetLtxMacro\LWR@origcapitalring\capitalring
28 \LetLtxMacro\LWR@origcapitalcaron\capitalcaron
29 \LetLtxMacro\LWR@origcapitalbreve\capitalbreve
30 \LetLtxMacro\LWR@origcapitalmacron\capitalmacron

```

```
31 \LetLtxMacro\LWR@origcapitaldotaccent\capitaldotaccent
32 \LetLtxMacro\LWR@origtextcircled\textcircled
```

For HTML, use HTML entities or direct Unicode, depending on the engine.

`\AtBeginDocument` improves support for Lua \TeX and Xe \TeX .

```
33 \AtBeginDocument{
34 \ifPDFTeX
35 \renewcommand*\textdegree{\HTMLentity{deg}}
36 \renewcommand*\textcelsius{\HTMLunicode{2103}}
37 \renewcommand*\textohm{\HTMLunicode{2126}}
38 \renewcommand*\textmu{\HTMLunicode{00B5}}
39 \renewcommand*\textlquill{\HTMLunicode{2045}}
40 \renewcommand*\textrquill{\HTMLunicode{2046}}
41 \renewcommand*\textcircledP{\HTMLunicode{2117}}
42 \renewcommand*\texttwelveudash{\HTMLunicode{2014}}% emdash
43 \renewcommand*\textthreequartersemdash{\HTMLunicode{2014}}% emdash
44 \renewcommand*\textmho{\HTMLunicode{2127}}
45 \renewcommand*\textnaira{\HTMLunicode{20A6}}
46 \renewcommand*\textpeso{\HTMLunicode{20B1}}
47 \renewcommand*\textrecipe{\HTMLunicode{211E}}
48 \renewcommand*\textpertenthousand{\HTMLunicode{2031}}
49 \renewcommand*\textbaht{\HTMLunicode{0E3F}}
50 \renewcommand*\textdiscount{\}%
51 \renewcommand*\textservicemark{\HTMLunicode{2120}}
52 \else
```

NOTE: Some of the following do not print well in the listing. Consult the .dtx or .sty file for the actual characters.

```
53 \renewcommand*\textdegree{\textdegree}
54 \renewcommand*\textcelsius{\textcelsius}
55 \renewcommand*\textohm{\textohm}
56 \renewcommand*\textmu{\textmu}
57 \renewcommand*\textlquill{\textlquill}
58 \renewcommand*\textrquill{\textrquill}
59 \renewcommand*\textcircledP{\textcircledP}
60 \renewcommand*\texttwelveudash{\texttwelveudash}
61 \renewcommand*\textthreequartersemdash{\textthreequartersemdash}
62 \renewcommand*\textmho{\textmho}
63 \renewcommand*\textnaira{\textnaira}
64 \renewcommand*\textpeso{\textpeso}
65 \renewcommand*\textrecipe{\textrecipe}
66 \renewcommand*\textpertenthousand{\textpertenthousand}
67 \renewcommand*\textbaht{\textbaht}
68 \renewcommand*\textdiscount{\textdiscount}
69 \renewcommand*\textservicemark{\textservicemark}
70 \fi
```

For HTML, Unicode diacritical marks are used:

```

71 \renewcommand*\capitalcedilla}[1]{#1\HTMLUnicode{0327}}
72 \renewcommand*\capitalogonek}[1]{#1\HTMLUnicode{0328}}
73 \renewcommand*\capitalgrave}[1]{#1\HTMLUnicode{0300}}
74 \renewcommand*\capitalacute}[1]{#1\HTMLUnicode{0301}}
75 \renewcommand*\capitalcircumflex}[1]{#1\HTMLUnicode{0302}}
76 \renewcommand*\capitaltilde}[1]{#1\HTMLUnicode{0303}}
77 \renewcommand*\capitaldieresis}[1]{#1\HTMLUnicode{0308}}
78 \renewcommand*\capitalhungarumlaut}[1]{#1\HTMLUnicode{30B}}
79 \renewcommand*\capitalring}[1]{#1\HTMLUnicode{30A}}
80 \renewcommand*\capitalcaron}[1]{#1\HTMLUnicode{30C}}
81 \renewcommand*\capitalbreve}[1]{#1\HTMLUnicode{306}}
82 \renewcommand*\capitalmacron}[1]{#1\HTMLUnicode{304}}
83 \renewcommand*\capitaldotaccent}[1]{#1\HTMLUnicode{307}}

```

`\textcircled` becomes a span with a rounded border:

```

84 \renewcommand*\textcircled}[1]{%
85 \InlineClass[border: 1px solid \LWR@currenttextcolor]{textcircled}{#1}%
86 }
87 }% AtBeginDocument

```

When a `lateximage` is begun:

```

88 \appto{\LWR@restoreorigformatting}{%
89 \let\textdegree\LWR@origtextdegree%
90 \let\textcelsius\LWR@origtextcelsius%
91 \let\textohm\LWR@origtextohm%
92 \let\textmu\LWR@origtextmu%
93 \let\textlquill\LWR@origtextlquill%
94 \let\textrquill\LWR@origtextrquill%
95 \let\textcircledP\LWR@origtextcircledP%
96 \let\texttwelveudash\LWR@origtexttwelveudash%
97 \let\textthreequartersemdash\LWR@origtextthreequartersemdash%
98 \let\textmho\LWR@origtextmho%
99 \let\textnaira\LWR@origtextnaira%
100 \let\textpeso\LWR@origtextpeso%
101 \let\textrecipe\LWR@origtextrecipe%
102 \let\textpertenthousand\LWR@origtextpertenthousand%
103 \let\textbaht\LWR@origtextbaht%
104 \let\textdiscount\LWR@origtextdiscount%
105 \let\textservicemark\LWR@origtextservicemark%
106 \LetLtxMacro\capitalcedilla\LWR@origcapitalcedilla%
107 \LetLtxMacro\capitalogonek\LWR@origcapitalogonek%
108 \LetLtxMacro\capitalgrave\LWR@origcapitalgrave%
109 \LetLtxMacro\capitalacute\LWR@origcapitalacute%
110 \LetLtxMacro\capitalcircumflex\LWR@origcapitalcircumflex%
111 \LetLtxMacro\capitaltilde\LWR@origcapitaltilde%

```

```

112 \LetLtxMacro\capitaldieresis\LWR@origcapitaldieresis%
113 \LetLtxMacro\capitalhungarumlaut\LWR@origcapitalhungarumlaut%
114 \LetLtxMacro\capitalring\LWR@origcapitalring%
115 \LetLtxMacro\capitalcaron\LWR@origcapitalcaron%
116 \LetLtxMacro\capitalbreve\LWR@origcapitalbreve%
117 \LetLtxMacro\capitalmacron\LWR@origcapitalmacron%
118 \LetLtxMacro\capitaldotaccent\LWR@origcapitaldotaccent%
119 \LetLtxMacro\textcircled\LWR@origtextcircled%
120 }

```

File 144 **lwarp-textpos.sty**

§ 220 Package **textpos**

Pkg textpos textpos is emulated during HTML output, and the textpos package is ignored.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{textpos}

2 \NewDocumentEnvironment{textblock}{m r()}{}{}
3 \NewDocumentEnvironment{textblock*}{m o r()}{}{}
4 \newcommand*{\TPGrid}[3] [] {}
5 \NewDocumentCommand{\TPMargin}{s o}{}
6 \newcommand*{\textblockcolour}[1] {}
7 \newcommand*{\textblockrulecolour}[1] {}
8 \newcommand*{\textblockcolor}[1] {}
9 \newcommand*{\textblockrulecolor}[1] {}
10 \newcommand*{\tekstblokkulur}[1] {}
11 \newcommand*{\tekstblokrulekulur}[1] {}
12 \newlength{\TPHorizModule}
13 \newlength{\TPVertModule}
14 \newlength{\TPboxrulesize}
15 \newcommand{\textblocklabel}[1] {}
16 \newcommand*{\showtextsize}{}
17 \newcommand{\textblockorigin}[2] {}

```

File 145 **lwarp-theorem.sty**

§ 221 Package **theorem**

(Based on original code by FRANK MITTELBACH.)

Pkg theorem theorem is patched for use by lwarp.

CSS styling of theorems and proofs:

Theorem: <div> of class theorembody<theoremstyle>

Theorem Header: of class theoremheader

where <theoremstyle> is plain, break, etc.

for HTML output: 1 \LWR@ProvidesPackagePass{theorem}

Storage for the style being used for new theorems:

```
2 \newcommand{\LWR@newtheoremstyle}{plain}
```

Patched to remember the style being used for new theorems:

```
3 \gdef\theoremstyle#1{%
4   \@ifundefined{th@#1}{\@warning
5     {Unknown theoremstyle '#1'. Using 'plain'}}%
6     \theorem@style{plain}%
7     \renewcommand{\LWR@newtheoremstyle}{plain}% new
8     }%
9   {%
10    \theorem@style{#1}%
11    \renewcommand{\LWR@newtheoremstyle}{#1}% new
12  }%
13  \begingroup
14    \csname th@the\theorem@style \endcsname
15  \endgroup}
```

Patched to remember the style for this theorem type, and set it later when the environment is started.

```
16 \gdef\xnthm#1#2[#3]{%
17   \expandafter\@ifdefinable\csname #1\endcsname
18   {%
19     \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
20     \@definecounter{#1}\@newctr{#1}[#3]%
21     \expandafter\xdef\csname the#1\endcsname
22     {\expandafter \noexpand \csname the#3\endcsname
23       \@thmcountersep \@thmcounter{#1}}%
24     \def\@tempa{\global\@namedef{#1}}%
25     \expandafter \@tempa \expandafter{%
26       \csname th@the \theorem@style
27         \expandafter \endcsname \the \theorem@bodyfont
28       \@thm{#1}{#2}}%
29     \global \expandafter \let \csname end#1\endcsname \@endtheorem
30     \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
31   }}
32
33 \gdef\@ynthm#1#2{%
```

```

34 \expandafter\@ifdefinable\csname #1\endcsname
35 {
36 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
37 \@definecounter{#1}%
38 \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
39 \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
40 \expandafter{\csname th@\the \theorem@style \expandafter
41 \endcsname \the\theorem@bodyfont \@thm{#1}{#2}}%
42 \global \expandafter \let \csname end#1\endcsname \@endtheorem
43 \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
44 }}
45
46 \gdef\@othm#1[#2]#3{%
47 \expandafter\ifx\csname c@#2\endcsname\relax
48 \nocounterr{#2}%
49 \else
50 \expandafter\@ifdefinable\csname #1\endcsname
51 {
52 \csedef{LWR@thmstyle#1}{\LWR@newtheoremstyle}% new
53 \expandafter \xdef \csname the#1\endcsname
54 {\expandafter \noexpand \csname the#2\endcsname}%
55 \def\@tempa{\global\@namedef{#1}}\expandafter \@tempa
56 \expandafter{\csname th@\the \theorem@style \expandafter
57 \endcsname \the\theorem@bodyfont \@thm{#2}{#3}}%
58 \global \expandafter \let \csname end#1\endcsname \@endtheorem
59 \AtBeginEnvironment{#1}{\edef\LWR@thisthmstyle{\csuse{LWR@thmstyle#1}}}% new
60 }%
61 \fi}

```

The following are patched for css.

These were in individual files thp.sty for plain, thmb.sty for margin break, etc. They are gathered together here.

Each theorem is encased in a BlockClass environment of class theorembody<style>.

Each header is encased in an \InlineClass of class theoremheader.

```

62 \gdef\th@plain{\normalfont\itshape
63 \def\@begintheorem##1##2{%
64 \LWR@forcenewpage% new
65 \BlockClass{theorembody\LWR@thisthmstyle}% new
66 \item[\hskip\labelsep
67 \InlineClass{theoremheader}{##1\ ##2}
68 ]}%
69 \def\@opargbegintheorem##1##2##3{%
70 \LWR@forcenewpage% new
71 \BlockClass{theorembody\LWR@thisthmstyle}% new
72 \item[\hskip\labelsep
73 \InlineClass{theoremheader}{##1\ ##2\ (##3)}

```

```

74   ]}
75 }
76
77 \gdef\th@break{%\normalfont\slshape
78   \def\@begintheorem##1##2{%
79 \LWR@forcenewpage% new
80   \BlockClass{theorembody\LWR@thisthmstyle}% new
81   \item[\hskip \labelsep
82     \InlineClass{theoremheader}{##1\ ##2}\newline%
83   ]}%
84 \def\@opargbegintheorem##1##2##3{%
85 \LWR@forcenewpage% new
86   \BlockClass{theorembody\LWR@thisthmstyle}% new
87   \item[\hskip \labelsep
88     \InlineClass{theoremheader}{##1\ ##2\ (##3)}\newline
89   ]}
90 }
91
92 \gdef\th@marginbreak{%\normalfont\slshape
93   \def\@begintheorem##1##2{
94 \LWR@forcenewpage% new
95   \BlockClass{theorembody\LWR@thisthmstyle}% new
96   \item[\hskip\labelsep %
97     \InlineClass{theoremheader}{##2 \quad ##1}\newline
98   ]}%
99 \def\@opargbegintheorem##1##2##3{%
100 \LWR@forcenewpage% new
101   \BlockClass{theorembody\LWR@thisthmstyle}% new
102   \item[\hskip\labelsep %
103     \InlineClass{theoremheader}{##2 \quad ##1\ %
104     (##3)}\newline
105   ]}
106 }
107
108 \gdef\th@changebreak{%\normalfont\slshape
109   \def\@begintheorem##1##2{
110 \LWR@forcenewpage% new
111   \BlockClass{theorembody\LWR@thisthmstyle}% new
112   \item[\hskip\labelsep
113     \InlineClass{theoremheader}{##2\ ##1}\newline
114   ]}%
115 \def\@opargbegintheorem##1##2##3{%
116 \LWR@forcenewpage% new
117   \BlockClass{theorembody\LWR@thisthmstyle}% new
118   \item[\hskip\labelsep
119     \InlineClass{theoremheader}{ ##2\ ##1\ %
120     (##3)}\newline
121   ]}
122 }
123

```

```

124 \gdef\th@change{%\normalfont\slshape
125 \def\@begintheorem##1##2{
126 \LWR@forcenewpage% new
127 \BlockClass{theorembody\LWR@thisthmstyle}% new
128 \item[\hskip\labelsep
129 \InlineClass{theoremheader}{##2\ ##1}
130 ]}%
131 \def\@opargbegintheorem##1##2##3{%
132 \LWR@forcenewpage% new
133 \BlockClass{theorembody\LWR@thisthmstyle}% new
134 \item[\hskip\labelsep
135 \InlineClass{theoremheader}{##2\ ##1\ (##3)}
136 ]}
137 }
138
139 \gdef\th@margin{%\normalfont\slshape
140 \def\@begintheorem##1##2{
141 \LWR@forcenewpage% new
142 \BlockClass{theorembody\LWR@thisthmstyle}% new
143 \item[\hskip\labelsep
144 \InlineClass{theoremheader}{##2 \quad ##1}
145 ]}%
146 \def\@opargbegintheorem##1##2##3{%
147 \LWR@forcenewpage% new
148 \BlockClass{theorembody\LWR@thisthmstyle}% new
149 \item[\hskip\labelsep
150 \InlineClass{theoremheader}{##2 \quad ##1\ (##3)}
151 ]}
152 }

Patched for CSS:

153 \gdef\@endtheorem{\endBlockClass\endtrivlist}

```

File 146 **lwarp-threeparttable.sty**

§ 222 Package **threeparttable**

Pkg **threeparttable** threeparttable is emulated during HTML output, and the threeparttable package is ignored.

Table notes are contained inside a CSS `<div>` of class `tnotes`. If `enumitem` is used, the note item labels are also individually highlighted with an additional CSS `` of class `tnoteitemheader`, otherwise they are plain text.

for HTML output: `1 \LWR@ProvidesPackageDrop{threeparttable}`

```

\LWR@printtablenote  <math>\langle text \rangle</math>
Prints the table note item header inside a CSS class of tnoteitemheader.
2 \newcommand{\LWR@printtablenote}[1]{\InlineClass{tnoteitemheader}{#1}}

Env  threeparttable  [<math>\langle alignment \rangle</math>] To emulate threeparttable:
3 \newenvironment*{threeparttable}[1][b]{-}{-}

Env  tablenotes      [<math>\langle options \rangle</math>]
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  <math>\langle text \rangle</math>
17 \newcommand{\tnote}[1]{\LWR@htmlspan{sup}{#1}}

```

File 147 **lwarp-tikz.sty**

§ 223 Package **tikz**

Pkg tikz tikz is supported.

Accept all options for lwarp-tikz:

```
1 \LWR@ProvidesPackagePass{tikz}
```

catcodes lwarp changes the catcode of \$ for its own use. The Tikz babel library temporarily changes catcodes back to normal for Tikz's use. tikz v3.0.0 introduced the babel library which handles catcode changes. For older versions, lwarp must change \$'s catcode itself.

Also see:

<https://tex.stackexchange.com/questions/16199/test-if-a-package-or-package-option-is-loaded>

```

for HTML output: 2 \begin{warpHTML}

3 \newboolean{LWR@tikzbabel}
4
5 \@ifpackagelater{tikz}{2013/12/20}% Test for Tikz version v3.0.0
6 {\usetikzlibrary{babel}\booltrue{LWR@tikzbabel}}
7 {\boolfalse{LWR@tikzbabel}}

```

Env `tikzpicture` `tikzpicture` environment is enclosed inside a `\lateximage`. May be used as-is, and its contents will be converted to an image.

```

8 \BeforeBeginEnvironment{tikzpicture}{%
9 \lateximage%
10 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
11 {}%
12 {\catcode'\$=3} % dollar sign is math shift
13 }
14
15 \AfterEndEnvironment{tikzpicture}{%
16 \endlateximage%
17 \ifbool{LWR@tikzbabel}% Test for Tikz version v3.0.0
18 {}%
19 {\catcode'\$=\active}%
20 }

21 \end{warpHTML}

```

File 148 `lwarp-titles.sty`

§ 224 Package **titles**

Pkg `titles` `titles` is loaded and used by `lwarp` during HTML output. All user options and macros are ignored and disabled.

Discard all options for `lwarp-titles`:

```

for HTML output: 1 \LWR@ProvidesPackageDrop{titles}

```

`\pagestyle` and `\thispagestyle` are already disabled in the `lwarp` code.

```

\newpagestyle {<name>} [<style>] {<commands>}

```

```

\renewpagestyle  {<name>} [<style>] {<commands>}
                2 \NewDocumentCommand{\renewpagestyle}{m o m}{}
                3 \NewDocumentCommand{\renewpagestyle}{m o m}{}

\sethead        [<el>] [<ec>] [<er>] {<ol>} {<oc>} {<or>}
\setfoot        [<el>] [<ec>] [<er>] {<ol>} {<oc>} {<or>}
                4 \NewDocumentCommand{\sethead}{o o o m m m}{}
                5 \NewDocumentCommand{\setfoot}{o o o m m m}{}

\setttitlemarks * {<names>}
                6 \NewDocumentCommand{\setttitlemarks}{s m}{}

\headrule
\footrule
                7 \newcommand*{\headrule}{}
                8 \newcommand*{\footrule}{}

\setheadrule    {<length>}
\setfootrule    {<length>}
                9 \newcommand*{\setheadrule}[1]{}
               10 \newcommand*{\setfootrule}[1]{}

\makeheadrule
\makefootrule
                11 \newcommand*{\makeheadrule}{}
                12 \newcommand*{\makefootrule}{}

\setmarkboth    {<code>}
                13 \newcommand{\setmarkboth}[1]{}

\widenhead
                14 \NewDocumentCommand{\widenhead}{s o o m m}{}

\botttitlemarks
\topttitlemarks
\firsttitlemarks
\nexttitlemarks

```

\outertitlemarks

\innertitlemarks

```
15 \newcommand*\bottitlemarks{}
16 \newcommand*\toptitlemarks{}
17 \newcommand*\firsttitlemarks{}
18 \newcommand*\nexttoptitlemarks{}
19 \newcommand*\outertitlemarks{}
20 \newcommand*\innertitlemarks{}
```

\newtitlemark * {<name>}

```
21 \NewDocumentCommand{\newtitlemark}{s m}{}

```

\pretitlemark * {<section>} {<text>}

```
22 \NewDocumentCommand{\pretitlemark}{s m m}{}

```

\ifsamemark {<group>} {<command>} {<true>} {<false>}

```
23 \newcommand{\ifsamemark}[4]{}

```

\setfloathead * [(<.>)] [(<.>)] [(<.>)] {(<.>)} {(<.>)} {(<.>)} {<extra>} [(<which>)]

\setfloatfoot * [(<.>)] [(<.>)] [(<.>)] {(<.>)} {(<.>)} {(<.>)} {<extra>} [(<which>)]

```
24 \NewDocumentCommand{\setfloathead}{s o o o m m m m m}{}

```

```
25 \NewDocumentCommand{\setfloatfoot}{s o o o m m m m m}{}

```

\nextfloathead * [(<.>)] [(<.>)] [(<.>)] {(<.>)} {(<.>)} {(<.>)} {<extra>} [(<which>)]

\nextfloatfoot * [(<.>)] [(<.>)] [(<.>)] {(<.>)} {(<.>)} {(<.>)} {<extra>} [(<which>)]

```
26 \NewDocumentCommand{\nextfloathead}{s o o o m m m m m}{}

```

```
27 \NewDocumentCommand{\nextfloatfoot}{s o o o m m m m m}{}

```

\newmarkset {<markset>}

```
28 \newcommand{\newmarkset}[1]{}

```

\newextramark * {<markset>} {<macro-name>}

```
29 \NewDocumentCommand{\newextramarkset}{s m m}{}

```

\botextramarks {<markset>}

\topextramarks {<markset>}

\firstextramarks {<markset>}

```

\nextextramarks   {\langle markset \rangle}
\outereextramarks {\langle markset \rangle}
\innereextramarks {\langle markset \rangle}

30 \newcommand{\botextramarks}[1]{}
31 \newcommand{\topextramarks}[1]{}
32 \newcommand{\firstextramarks}[1]{}
33 \newcommand{\nexttopextramarks}[1]{}
34 \newcommand{\outereextramarks}[1]{}
35 \newcommand{\innereextramarks}[1]{}

```

File 149 **lwarp-titleref.sty**

§ 225 Package **titleref**

Pkg titleref titleref is superseded by hyperref and nameref.

for HTML output: 1 \LWR@loadnever{titleref}{hyperref and nameref}

File 150 **lwarp-titlesec.sty**

§ 226 Package **titlesec**

Pkg titlesec titlesec is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titlesec:

for HTML output: 1 \LWR@ProvidesPackageDrop{titlesec}

```

\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 * {<command>} {<left>} {<before>} {<after>} [<right>]
      10 \NewDocumentCommand{\titlespacing}{s m m m m o}{}

\filright
\filcenter
\filleft
\fillast
\filinner
\filouter
      11 \newcommand*\filright{}
      12 \newcommand*\filcenter{}
      13 \newcommand*\filleft{}
      14 \newcommand*\fillast{}
      15 \newcommand*\filinner{}
      16 \newcommand*\filouter{}

\wordsep
      17 \newcommand\wordsep{\fontdimen\tw@\font \@plus
      18 \fontdimen\thr@@\font \@minus \fontdimen4\font}

\titleline * [<align>] {<material>}
      19 \NewDocumentCommand{\titleline}{s o m}{}

\titlerule [<height>]
      20 \providecommand\titlerule{\@ifstar{\ttl@row}{\ttl@rule}}
      21 \newcommand*\ttl@rule[1] [] {}
      22 \newcommand*\ttl@row[2] [] {}

\iftitlemeasuring {<true>} {<false>}
\assignpagestyle {<command>} {<pagestyle>}
\titleclass {<name>} [<startlevel>] {<class>} [<cmd>]
      23 \newcommand\iftitlemeasuring[2] {#2}
      24 \newcommand\assignpagestyle[2] {#2}
      25 \NewDocumentCommand\titleclass{m o m o}

```

File 151 **lwarp-titletoc.sty**

§ 227 Package **titletoc**

Pkg titletoc titletoc is emulated. All user options and macros are ignored and disabled.

Discard all options for lwarp-titletoc:

for HTML output: 1 \LWR@ProvidesPackageDrop{titletoc}

\dottedcontents {<section>} [<left>] {<above>} {<label>} {<leader>}
 2 \NewDocumentCommand{\dottedcontents}{m o m m m}{}

\titlecontents * {<section>} [<left>] {<above>} {<numbered>} {<numberless>} {<filler>} [<below
 or begin>] [<separator>] [<end>]
 3 \newcommand{\titlecontents}{\ifstar{\ttl@tcstar}{\ttl@tcnostar}}
 4 \NewDocumentCommand{\ttl@tcstar}{m o m m m o o}{}
 5 \NewDocumentCommand{\ttl@tcnostar}{m o m m m m o}{}

\contentsmargin [<correction>] {<right>}
 6 \newcommand{\contentsmargin}[2] [] {}

\thecontentslabel
 \thecontentspage
 7 \newcommand*{\thecontentslabel}{thecontentslabel}
 8 \newcommand*{\thecontentspage}{thecontentspage}

\contentslabel [<format>] {<space>}
 \contentspage [<format>]
 9 \newcommand{\contentslabel}[2] [] {\thecontentslabel}
 10 \newcommand{\contentspage}[1] [] {\thecontentspage}

\contentspush {<text>}
 11 \newcommand{\contentspush}[1] {}

\contentsuse {<name>} {<text>}

```

12 \newcommand{\contentsuse}[2]{}

\startcontents [name]
\stopcontents  [name]
\resumecontents [name]

13 \newcommand*{\startcontents}[1] [] {}
14 \newcommand*{\stopcontents}[1] [] {}
15 \newcommand*{\resumecontents}[1] [] {}

\printcontents [name] {prefix} {start} {code}

16 \newcommand{\printcontents}[4] [] {}

\startlist    [name] {list}
\stoplist     [name] {list}
\resumelist   [name] {list}

17 \newcommand{\startlist}[2] [] {}
18 \newcommand{\stoplist}[2] [] {}
19 \newcommand{\resumelist}[2] [] {}

\printlist    [name] {list} {prefix} {code}

20 \newcommand{\printlist}[4] [] {}

```

File 152 **lwarp-titling.sty**

§ 228 Package **titling**

Pkg titling

package support lwarp supports the native L^AT_EX titling commands, and also supports the packages authblk and titling. If both are used, authblk should be loaded before titling.

 **load order**

\published and \subtitle If using the titling package, additional titlepage fields for `\published` and `\subtitle` may be added by using `\AddSubtitlePublished` in the preamble. See section 52.7.

The various titling footnote restyling commands have no effect.

Pass all options to `lwarp-titling`:

for HTML output: `1 \LWR@ProvidesPackagePass{titling}`

`\@bsmtitleempty` Patch `\@bsmtitleempty`:

```

2 \let\LWR@orig@bsmtitleempty\@bsmtitleempty
3 \renewcommand*\@bsmtitleempty}{%
4 \LWR@orig@bsmtitleempty%
5 }

```

`\keepthetitle` Patch `\keepthetitle`:

```

6 \let\LWR@origkeepthetitle\keepthetitle
7 \renewcommand*\keepthetitle}{%
8 \LWR@orig@keepthetitle%
9 }

```

`\killtitle` Patch `\killtitle`:

```

10 \let\LWR@origkilltitle\killtitle
11 \renewcommand*\killtitle}{%
12 \LWR@orig@killtitle%
13 }

```

Env `titlingpage`

```

14 \renewenvironment*{titlingpage}
15 {%

```

Start an HTML titlepage div:

```

16 \LWR@printpendingfootnotes
17 \begin{titlepage}

```

Prepare for a custom version of `\maketitle` inside the `titlingpage`:

```

18 \LWR@maketitlesetup
19 \let\maketitle\LWR@titlingmaketitle
20 }
21 {

```

At the end of the environment, end the HTML titlepage div:

```

22 \end{titlepage}
23 }

```

Patch the pre/post title/author/date to add HTML tags, then initialize:

```

24
25 \pretitle{}
26 \posttitle{}
27

```

```

28 \preauthor{}
29 \postauthor{}
30
31 \predate{}
32 \postdate{}

```

`\LWR@maketitlesetup` Patches `\thanks` macros.

```
33 \renewcommand*{\LWR@maketitlesetup}{%
```

Redefine the footnote mark:

```
34 \def\@makefnmark{\textsuperscript{\@thefnmark}}
```

```

\thefootnote ⇒ \nameuse{arabic}{footnote}, or
\thefootnote ⇒ \nameuse{fnsymbol}{footnote}

```

Redefine the footnote text:

```
35 \long\def\@makefntext##1{%
```

Make the footnote mark and some extra horizontal space for the tags:

```
36 \makethanksmark \LWR@origspace{1in}
```

```

\makethanksmark ⇒ \thanksfootmark ⇒ \tamark ⇒
\@thefnmark ⇒ \itshape a (or similar)

```

Print the text:

```

37 ##1%
38 }%
39 }

```

`\maketitle` HTML mode. Creates an HTML titlepage div and typesets the title, etc.

Code from the titling package is adapted, simplified, and modified for HTML output.

```
40 \renewcommand*{\maketitle}{%
```

An HTML titlepage `<div>` is used for all classes.

```
41 \begin{titlepage}
```

Select which kind of footnote marks to use:

```
42 \@bsmarkseries
```

Set up special patches:

```
43 \LWR@maketitlesetup
```

Typeset the title, etc:

```
44 \@maketitle
```

Immediately generate any \thanks footnotes:

```
45 \@thanks
```

Close the HTML titlepage div:

```
46 \end{titlepage}
```

Reset the footnote counter:

```
47 \@bscontmark
48 }
```

`\@maketitle` Typesets the title, etc. Patched for HTML.

```
49 \DeclareDocumentCommand{\@maketitle}{-}{%
50   \maketitlehooka
51   {
52     \LWR@stoppars\LWR@htmltag{h1}
53     \@bsprefix \@title \@bsprefix
54     \LWR@htmltag{/h1}\LWR@startpars
55   }
56   \maketitlehookb
57   {
58     \begin{BlockClass}{author}
59     \renewcommand{\and}{
60       \end{BlockClass}
61       \begin{BlockClass}{oneauthor}
62     }
63     \begin{BlockClass}{oneauthor}
64     \@bsprefix \@author \@bsprefix
65     \end{BlockClass}
66     \end{BlockClass}
67   }
68   \maketitlehookc
69   {
```

```

70     \begin{BlockClass}{titledate}
71     \@bspredate \@date \@bspostdate
72     \end{BlockClass}
73   }
74   \maketitlehookd
75 }

```

`\LWR@titlingmaketitle` `\maketitle` for use inside an HTML titlingpage environment.

```
76 \renewcommand*{\LWR@titlingmaketitle}{%
```

Keep pending footnotes out of the title block:

```
77 \@thanks
```

Select which kind of footnote marks to use:

```
78 \@bsmarkseries
```

Set up special patches:

```
79 \LWR@maketitlesetup
```

Typeset the title, etc:

```
80 \@maketitle
```

Immediately generate any `\thanks` footnotes:

```
81 \@thanks
```

Reset the footnote counter:

```
82 \@bscontmark
83 }
```

`\thanksmarkseries` `{\langle series \rangle}`

Sets the type of footnote marks used by `\thanks`, where type is ‘arabic’, ‘roman’, ‘fnsymbol’, etc.

```
84 \renewcommand{\thanksmarkseries}[1]{%
85 \def\@bsmarkseries{\renewcommand{\thefootnote}{\@nameuse{#1}{footnote}}}%
86 }
```

Set default titlepage thanks footnote marks. See section [52.6](#).

```

87 \if@titlepage
88   \thanksmarkseries{arabic}
89 \else
90   \thanksmarkseries{fnsymbol}
91 \fi

```

File 153 **lwarp-tocbibind.sty**

§ 229 Package **tocbibind**

Pkg tocbibind tocbibind is patched for use by lwarp.

Opt IndexLanguage The lwarp package takes an option `IndexLanguage=english` to set the language used by xindy. This is passed to xindy using its `-L` option, and is used for both index and glossary generation.

 **tocloft & other packages** If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard \LaTeX commands to create the titles, allowing other packages to work with it.

placement and roc options An index may be placed inline with other HTML text, or on its own HTML page:

Inline, with a manual TOC entry:

A commonly-used method to introduce an index in a \LaTeX document:

```

\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\printindex

```

On its own HTML page, with a manual TOC entry:

```

\begin{warpprint}
\cleardoublepage
\phantomsection
\addcontentsline{toc}{section}{\indexname}% or chapter
\end{warpprint}
\ForceHTMLPage
\ForceHTMLTOC
\printindex

```

Inline, with an automatic TOC entry:

Pkg tocbibind The tocbibind package may be used to automatically place an entry in the toc.

```

\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\printindex

```

On its own HTML page, with an automatic TOC entry:

```

\usepackage[nottoc]{tocbibind}
...
\cleardoublepage
\phantomsection % to fix print-version index link
\ForceHTMLPage
\printindex

```

`Opt tocbibind numindex` Use the `tocbibind numindex` option to generate a numbered index. Without this option, the index heading has no number.

numbered index section

for HTML output:

```

1 \let\simplechapterdelim\relax
2
3 \LWR@ProvidesPackagePass{tocbibind}

4 \renewenvironment{theindex}%
5   {%
6     \if@bibchapter
7       \if@donumindex
8         \chapter{\indexname}
9       \else
10      \if@dotocind
11        \chapter*{\indexname}
12        \addcontentsline{toc}{chapter}{\indexname}
13      \else
14        \chapter*{\indexname}
15      \fi
16    \fi
17  \else
18    \if@donumindex
19      \section{\indexname}
20    \else
21      \if@dotocind
22        \section*{\indexname}
23        \addcontentsline{toc}{\@tocextra}{\indexname}
24      \else
25        \section*{\indexname}
26      \fi
27    \fi
28  \fi
29 \let\item\LWR@indexitem%
30 \let\subitem\LWR@indexsubitem%
31 \let\subsubitem\LWR@indexsubsubitem%

```

```
32 }
```

The following code is shared by anonchap.

```
33 \renewcommand{\simplechapter}[1][\@empty]{%
34   \def\@chapcntformat##1{%
35     #1~\csname the##1\endcsname\simplechapterdelim\protect\quad%
36   }%
37 }
38
39 \renewcommand{\restorechapter}{%
40 \let\@chapcntformat\@seccntformat%
41 }
```

File 154 **lwarp-tocloft.sty**

§ 230 Package **tocloft**

Pkg tocloft tocloft is emulated. Most user options and macros are ignored and disabled. `\newlistof` and `\cftchapterprecis` are supported.

 **tocloft & other packages** If using tocloft with tocbibind, anonchap, fncychap, or other packages which change chapter title formatting, load tocloft with its `titles` option, which tells tocloft to use standard \LaTeX commands to create the titles, allowing other packages to work with it.

Discard all options for lwarp-tocloft:

for HTML output: `1 \LWR@ProvidesPackageDrop{tocloft}`

`\tocloftpagestyle` `{\style}`
`2 \newcommand{\tocloftpagestyle}[1]{}`

`\cftmarktoc`
`\cfttoctitlefont`
`\cftaftertocitle`
`3 \newcommand*\cftmarktoc{}`
`4 \newcommand*\cfttoctitlefont{}`
`5 \newcommand*\cftaftertocitle{}`
`6 \newlength{\cftbeforetoctitleskip}`
`7 \newlength{\cftaftertocitleskip}`

```

\cftmarklof
\cftlofttitlefont
\cftafterlofttitle

8 \newcommand*{\cftmarklof}{}
9 \newcommand*{\cftlofttitlefont}{}
10 \newcommand*{\cftafterlofttitle}{}

11 \newlength{\cftbeforelofttitleskip}
12 \newlength{\cftafterlofttitleskip}

\cftmarklot
\cftlottitlefont
\cftafterlottitle

13 \newcommand*{\cftmarklot}{}
14 \newcommand*{\cftlottitlefont}{}
15 \newcommand*{\cftafterlottitle}{}

16 \newlength{\cftbeforelottitleskip}
17 \newlength{\cftafterlottitleskip}

\cftdot
\cftdotsep
\cftnodots

18 \newcommand*{\cftdot}{.}
19 \providecommand*{\cftdotsep}[1]{}
20 \newcommand*{\cftnodots}{5000}

\cftdotfill {sep}

21 \providecommand{\cftdotfill}[1]{}

\cftsetpnumwidth {length}
\cftsetrmarg {length}

22 \newcommand*{\cftsetpnumwidth}[1]{}
23 \newcommand*{\cftsetrmarg}[1]{}

\cftpnumalign {alignment}

24 \newcommand*{\cftpnumalign}[1]{}

25 \newlength{\cftparskip}

```

```
26 \newlength{\cftbeforepartskip}
27 \newlength{\cftpartindent}
28 \newlength{\cftpartnumwidth}
29 \newcommand*{\cftpartfont}{}
30 \newcommand*{\cftpartpresnum}{}
31 \newcommand*{\cftpartaftersnum}{}
32 \newcommand*{\cftpartaftersnumb}{}
33 \newcommand*{\cftpartleader}{}
34 \newcommand*{\cftpartdotsep}{1}
35 \newcommand*{\cftpartpagefont}{}
36 \newcommand*{\cftpartafterpnum}{}

37 \newlength{\cftbeforechapskip}
38 \newlength{\cftchapindent}
39 \newlength{\cftchapnumwidth}
40 \newcommand*{\cftchapfont}{}
41 \newcommand*{\cftchappresnum}{}
42 \newcommand*{\cftchapaftersnum}{}
43 \newcommand*{\cftchapaftersnumb}{}
44 \newcommand*{\cftchapleader}{}
45 \newcommand*{\cftchapdotsep}{1}
46 \newcommand*{\cftchappagefont}{}
47 \newcommand*{\cftchapafterpnum}{}

48 \newlength{\cftbeforesecskip}
49 \newlength{\cftsecindent}
50 \newlength{\cftsecnumwidth}
51 \newcommand*{\cftsecfont}{}
52 \newcommand*{\cftsecpresnum}{}
53 \newcommand*{\cftsecaftersnum}{}
54 \newcommand*{\cftsecaftersnumb}{}
55 \newcommand*{\cftsecleader}{}
56 \newcommand*{\cftsecdotsep}{1}
57 \newcommand*{\cftsecpagefont}{}
58 \newcommand*{\cftsecafterpnum}{}

59 \newlength{\cftbeforesubsecskip}
60 \newlength{\cftsubsecindent}
61 \newlength{\cftsubsecnumwidth}
62 \newcommand*{\cftsubsecfont}{}
63 \newcommand*{\cftsubsecpresnum}{}
64 \newcommand*{\cftsubsecaftersnum}{}
65 \newcommand*{\cftsubsecaftersnumb}{}
66 \newcommand*{\cftsubsecleader}{}
67 \newcommand*{\cftsubsecdotsep}{1}
68 \newcommand*{\cftsubsecpagefont}{}
69 \newcommand*{\cftsubsecafterpnum}{}

70 \newlength{\cftbeforesubsubsecskip}
```

```
71 \newlength{\cftsubsubsecindent}
72 \newlength{\cftsubsubsecnumwidth}
73 \newcommand*\cftsubsubsecfont-{}
74 \newcommand*\cftsubsubsecpresnum-{}
75 \newcommand*\cftsubsubsecaftersnum-{}
76 \newcommand*\cftsubsubsecaftersnumb-{}
77 \newcommand*\cftsubsubsecleader-{}
78 \newcommand*\cftsubsubsecdotsep-{}{1}
79 \newcommand*\cftsubsubsecpagefont-{}
80 \newcommand*\cftsubsubsecafterpnum-{}

81 \newlength{\cftbeforeparaskip}
82 \newlength{\cftparaindent}
83 \newlength{\cftparanumwidth}
84 \newcommand*\cftparafont-{}
85 \newcommand*\cftparapresnum-{}
86 \newcommand*\cftparaaftersnum-{}
87 \newcommand*\cftparaaftersnumb-{}
88 \newcommand*\cftparaleader-{}
89 \newcommand*\cftparadotsep-{}{1}
90 \newcommand*\cftparapagefont-{}
91 \newcommand*\cftparaafterpnum-{}

92 \newlength{\cftbeforesubparaskip}
93 \newlength{\cftsubparaindent}
94 \newlength{\cftsubparanumwidth}
95 \newcommand*\cftsubparafont-{}
96 \newcommand*\cftsubparapresnum-{}
97 \newcommand*\cftsubparaaftersnum-{}
98 \newcommand*\cftsubparaaftersnumb-{}
99 \newcommand*\cftsubparaleader-{}
100 \newcommand*\cftsubparadotsep-{}{1}
101 \newcommand*\cftsubparapagefont-{}
102 \newcommand*\cftsubparaafterpnum-{}

103 \newlength{\cftbeforefigskip}
104 \newlength{\cftfigindent}
105 \newlength{\cftfignumwidth}
106 \newcommand*\cftfigfont-{}
107 \newcommand*\cftfigpresnum-{}
108 \newcommand*\cftfigaftersnum-{}
109 \newcommand*\cftfigaftersnumb-{}
110 \newcommand*\cftfigleader-{}
111 \newcommand*\cftfigdotsep-{}{1}
112 \newcommand*\cftfigpagefont-{}
113 \newcommand*\cftfigafterpnum-{}

114 \newlength{\cftbeforesubfigskip}
115 \newlength{\cftsubfigindent}
```

```

116 \newlength{\cftsubfignumwidth}
117 \newcommand*{\cftsubfigfont}{}
118 \newcommand*{\cftsubfigpresnum}{}
119 \newcommand*{\cftsubfigaftersnum}{}
120 \newcommand*{\cftsubfigaftersnumb}{}
121 \newcommand*{\cftsubfigleader}{}
122 \newcommand*{\cftsubfigdotsep}{1}
123 \newcommand*{\cftsubfigpagefont}{}
124 \newcommand*{\cftsubfigafterpnum}{}

```

```

125 \newlength{\cftbeforetabskip}
126 \newlength{\cfttabindent}
127 \newlength{\cfttabnumwidth}
128 \newcommand*{\cfttabfont}{}
129 \newcommand*{\cfttabpresnum}{}
130 \newcommand*{\cfttabaftersnum}{}
131 \newcommand*{\cfttabaftersnumb}{}
132 \newcommand*{\cfttableader}{}
133 \newcommand*{\cfttabdotsep}{1}
134 \newcommand*{\cfttabpagefont}{}
135 \newcommand*{\cfttabafterpnum}{}

```

```

136 \newlength{\cftbeforesubtabskip}
137 \newlength{\cftsubtabindent}
138 \newlength{\cftsubtabnumwidth}
139 \newcommand*{\cftsubtabfont}{}
140 \newcommand*{\cftsubtabpresnum}{}
141 \newcommand*{\cftsubtabaftersnum}{}
142 \newcommand*{\cftsubtabaftersnumb}{}
143 \newcommand*{\cftsubtableader}{}
144 \newcommand*{\cftsubtabdotsep}{1}
145 \newcommand*{\cftsubtabpagefont}{}
146 \newcommand*{\cftsubtabafterpnum}{}

```

```

147 \newcommand{\cftsetindents}[3]{}

```

```

148 \newcommand{\pagenumbersoff}[1]{}

```

```

149 \newcommand{\pagenumberon}[1]{}

```

`\newlistof` [*<within>*] [*<type>*] [*<ext>*] [*<listofname>*]

Emulated through the `\newfloat` mechanism.

```

150 \NewDocumentCommand{\newlistof}{o m m m}
151 {%
152 \IfValueTF{#1}
153 {\newfloat{#2}{tbp}{#3}[#1]}
154 {\newfloat{#2}{tbp}{#3}}
155 \@namedef{listof#2}{\listof{#2}{#4}}

```

```

156 \@namedef{#2depth}{1}
157 \expandafter\newlength\csuse{cftbefore#2skip}
158 \expandafter\newlength\csuse{cft#2indent}
159 \expandafter\newlength\csuse{cft#2numwidth}
160 \@namedef{cft#2font}{}
161 \@namedef{cft#2presnum}{}
162 \@namedef{cft#2aftersnum}{}
163 \@namedef{cft#2aftersnumb}{}
164 \@namedef{cft#2leader}{}
165 \@namedef{cft#2dotsep}{1}
166 \@namedef{cft#2pagefont}{}
167 \@namedef{cft#2afterpnum}{}
168 }

```

\cftchapterprecis from tocloft:

```

169 \newcommand{\cftchapterprecis}[1]{%
170 \cftchapterprecishere{#1}
171 \cftchapterprecistoc{#1}}
172 \newcommand{\cftchapterprecishere}[1]{%
173 \begin{quote}\textit{#1}\end{quote}}
174 \newcommand{\cftchapterprecistoc}[1]{
175 \addtocontents{toc}{%
176 {
177 \protect\begin{quote}#1\protect\end{quote}}
178 }
179 }

```

File 155 **lwarp-transparent.sty**

§ 231 Package **transparent**

(Based on original code by HEIKO OBERDIEK.)

Pkg transparent Emulated. \texttransparent works for inline objects. \transparent only works for \includegraphics.

 **Not X_YL_AT_EX!** Note that transparent does not work with X_YL_AT_EX.

for HTML output: Discard all options for lwarp-transparent:

```

1 \LWR@ProvidesPackageDrop{transparent}

2 \newcommand*{\transparent}[1]{\edef\LWR@opacity{#1}}
3
4 \newcommand*{\texttransparent}[2]{%
5 \begingroup%

```

```

6 \transparent{#1}%
7 \InlineClass[opacity: #1]{transparent}{#2}%
8 \endgroup%
9 }

```

File 156 **lwarp-trivfloat.sty**

§ 232 Package **trivfloat**

Pkg trivfloat trivfloat is forced to use the built-in lwarp emulation for floats.

Discard all options for lwarp-trivfloat. This tells trivfloat not to use floatrow or memoir.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{trivfloat}
2 \LWR@origRequirePackage{trivfloat}

```

for HTML & PRINT:

```

3 \begin{warpall}

```

To create a new float type and change its name:

```

\trivfloat{example}
\renewcommand{\examplename}{Example Name}
\crefname{example}{example}{examples}
\Crefname{example}{Example}{Examples}

```

```

4 \end{warpall}

```

`\tfl@chapter@fix` Nullified at the beginning of the document. Is used by trivfloat to correct float chapter numbers, but is not needed for lwarp.

for HTML output:

```

5 \begin{warpHTML}

6 \AtBeginDocument{\DeclareDocumentCommand{\tfl@chapter@fix}{m m}{}}

7 \end{warpHTML}

```

§ 232.1 Combining `\newfloat`, `\trivfloat`, and `algorithmicx`

for HTML & PRINT: 8 `\begin{warpall}`

For both print and HTML output:

- ⚠ When using `float`, `trivfloat`, or `algorithmicx` at the same time, be aware of conflicting file usage. `algorithmicx` uses `.loa`. `trivfloat` by default starts with `.loa` and goes up for additional floats, skipping `.lof` and `.lot`.
- ⚠ When using `\newfloat`, be sure to manually assign higher letters to the `\newfloat` files to avoid `.loa` used by `algorithmicx`, and any files used by `trivfloat`. Also avoid using `.lof` and `.lot`.
- ⚠ When using `\trivfloat`, you may force it to avoid conflicting with `algorithmicx` by starting `trivfloat`'s file extensions with `.lob`:

```

\makeatletter
\setcounter{tfl@float@cnt}{1} % start trivfloats with .lob
\makeatletter

9 \end{warpall}

```

File 157 `lwarp-typearea.sty`

§ 233 Package **typearea**

Pkg `typearea` `typearea` is emulated.

for HTML output:

```

1 \LWR@ProvidesPackageDrop{typearea}

2 \newcommand*\typearea[2][\z@]{}
3 \newcommand*\recalc\typearea{}
4 \@ifundefined{footheight}{\newlength\footheight}{}
5 \newcommand*\areaset[3][\z@]{}

```

File 158 `lwarp-ulem.sty`

§ 234 Package **ulem**

(Based on original code by DONALD ARSENEAU.)

Pkg ulem Emulated.

for HTML output: Original lwarp definitions:

```
1 \LetLtxMacro\LWR@ulemorigemph\emph
2 \LetLtxMacro\LWR@ulemorigtextbf\textbf
```

Basic markup commands, using css:

```
3 \NewDocumentCommand{\uline}{+m}{%
4 \InlineClass{uline}{#1}%
5 }
6
7 \NewDocumentCommand{\uuline}{+m}{%
8 \InlineClass{uuline}{#1}%
9 }
10
11 \NewDocumentCommand{\uwave}{+m}{%
12 \InlineClass{uwave}{#1}%
13 }
14
15 \NewDocumentCommand{\sout}{+m}{%
16 \InlineClass{sout}{#1}%
17 }
18
19 \NewDocumentCommand{\xout}{+m}{%
20 \InlineClass{xout}{#1}%
21 }
22
23 \NewDocumentCommand{\dashuline}{+m}{%
24 \InlineClass{dashuline}{#1}%
25 }
26
27 \NewDocumentCommand{\dotuline}{+m}{%
28 \InlineClass{dotuline}{#1}%
29 }
```

Nullified parameters:

```
30 \NewDocumentCommand{\ULthickness}{-}{-}
31 \newlength{\ULdepth}
```

Nullified/emulated macros:

```
32 \NewDocumentCommand{\markoverwith}{m}{-}
33 \NewDocumentCommand{\ULon}{+m}{\uline{#1}\egroup}
```

\useunder only works with \textbf, etc, but not \bfseries, etc.

```

34 \NewDocumentCommand{\useunder}{m m m}{%
35 \relax%
36 \ifx\relax#3\relax\else % argumentative command
37   \def#3{#1}\MakeRobust{#3}\fi
38 }

```

Triggered by package options, also available for the users:

```

39 \newcommand*{\normalem}{\LetLtxMacro\emph\LWR@ulemorigemph}
40 \newcommand*{\ULforem}{\LetLtxMacro\emph\uline}
41 \ULforem% default

```

Package options:

```

42 \DeclareOption{normalem}{\normalem}
43 \DeclareOption{ULforem}{\ULforem}
44 \DeclareOption{normalbf}{}
45 \DeclareOption{UWforbf}{\useunder{\uwave}{\bf}{\textbf}}

```

Emulate the original package:

```

46 \LWR@ProvidesPackageDrop{ulem}

```

File 159 **lwarp-upref.sty**

§ 235 Package **upref**

Pkg upref Ignored.

for HTML output: Discard all options for lwarp-upref:

```

1 \LWR@ProvidesPackageDrop{upref}

```

File 160 **lwarp-verse.sty**

§ 236 Package **verse**

(Based on original code by PETER WILSON.)

Pkg verse verse is supported and patched by lwarp.

for HTML output: Pass all options for lwarp-verse:

```

1 \LWR@ProvidesPackagePass{verse}

```

`\attrib` The documentation for the `verse` and `memoir` packages suggest defining an `\attrib` command, which may already exist in current documents, but it will only work for print output. `lwarp` provides `\attribution`, which works for both print and HTML output. To combine the two so that `\attrib` is used for print and `\attribution` is used for HTML:

```
\begin{warpHTML}
\let\attrib\attribution
\end{warpHTML}
```

`Len \leftskip` These lengths are used by `verse` and `memoir` to control the left margin, and they may already be set by the user for print output. New lengths `\HTMLvleftskip` and `\HTMLleftmargini` are provided to control the margins in HTML output. These new lengths may be set by the user before any `verse` environment, and persist until they are manually changed again. One reason to change `\HTMLleftmargini` is if there is a wide `\flagverse` in use, such as the word “Chorus”, in which case the value of `\HTMLleftmargini` should be set to a wide enough length to contain “Chorus”. The default is wide enough for a stanza number.

`Len \leftmargini`

`Len \TMLvleftskip`

`Len \TMLleftmargini`

Horizontal spacing relies on `pdftotext`'s ability to discern the layout (`-layout` option) of the text in the HTML-tagged PDF output. For some settings of `\HTMLleftmargini` or `\HTMLleftskip` the horizontal alignment may not work out exactly, in which case a label may be shifted by one space.

`Env verse` The `verse` environment will be placed inside a HTML `pre`.

```
2 \AfterEndPreamble{
3 \LWR@traceinfo{Patching verse.}
```

At the beginning of the `verse` environment:

```
4 \AtBeginEnvironment{verse}
5 {%
```

`Pkg verse` The `verse` or `memoir` packages can place stanza numbers to the left with their `\flagverse` command. Do not allow them to go into the left margin, which would cause `pdfcrop` to crop the entire page further to the left:

`Pkg memoir`

`\flagverse`

```
Len \leftskip
6 \ifdef{\vleftskip}{%
7 \setlength{\vleftskip}{\HTMLvleftskip}
8 \setlength{\leftmargini}{\HTMLleftmargini}
9 }{}
10 \LWR@forcenewpage
```

```

11 \LWR@atbeginverbatim{verse}
12 \unskip\LWR@origvspace{-\baselineskip}
13 }

```

After the end of the verse environment, which places the pre tag at the regular left margin:

```

14 \AfterEndEnvironment{verse}{
15 \unskip\LWR@origvspace{-\baselineskip}
16 \LWR@afterendverbatim
17 }

```

Patch to place poemtitle inside an HTML span of class poemtitle:

```

18 \ifdef{\poemtitle}{
19 \DeclareDocumentCommand{\@vstypeptitle}{m}{%
20   \vspace{\beforepoemtitleskip}%
21   {\InlineClass{poemtitle}{\poemtitlefont #1}\par}%
22   \vspace{\afterpoemtitleskip}%
23 }
24 }{}
25
26 }% AfterEndPreamble

```

File 161 **lwarp-wallpaper.sty**

§ 237 Package **wallpaper**

Pkg wallpaper wallpaper is emulated during HTML output, and the wallpaper package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{wallpaper}

```

2 \newcommand*{\CenterWallPaper}[2]{}
3 \newcommand*{\ThisCenterWallPaper}[2]{}
4 \newcommand*{\TileWallPaper}[3]{}
5 \newcommand*{\ThisTileWallPaper}[3]{}
6 \newcommand*{\TileSquareWallPaper}[2]{}
7 \newcommand*{\ThisTileSquareWallPaper}[2]{}
8 \newcommand*{\ULCornerWallPaper}[2]{}
9 \newcommand*{\ThisULCornerWallPaper}[2]{}
10 \newcommand*{\LLCornerWallPaper}[2]{}
11 \newcommand*{\ThisLLCornerWallPaper}[2]{}
12 \newcommand*{\URCornerWallPaper}[2]{}
13 \newcommand*{\ThisURCornerWallPaper}[2]{}
14 \newcommand*{\LRCornerWallPaper}[2]{}
15 \newcommand*{\ThisLRCornerWallPaper}[2]{}

```

```

16 \newcommand*\ClearWallPaper{}
17 \newlength{\wpXoffset}
18 \newlength{\wpYoffset}

```

File 162 **lwarp-wrapfig.sty**

§ 238 Package **wrapfig**

Pkg wrapfig wrapfig is emulated during HTML output, and the wrapfig package is ignored.

for HTML output: 1 \LWR@ProvidesPackageDrop{wrapfig}

Computed width of a wrapped object. Used to print the HTML style.

```

2 \newlength{\LWR@wrapwidth}

3 \newcommand*\LWR@wrapposition{}
4
5 \newcommand*\LWR@subwrapfigure}[2]{%
6 \LWR@maybeinthisfloat%
7 \renewcommand*\LWR@wrapposition{}%
8 \ifthenelse{%
9   \equal{#1}{r}\OR\equal{#1}{R}\OR%
10  \equal{#1}{o}\OR\equal{#1}{O}%
11 }{%
12 {\renewcommand*\LWR@wrapposition}{float:right}}%
13 {\renewcommand*\LWR@wrapposition}{float:left}}%
14 \setlength{\LWR@wrapwidth}{#2}%
15 \addtolength{\LWR@wrapwidth}{4em}%
16 \uselengthunit{PT}%
17 \LWR@forcenewpage
18 \LWR@stoppars%
19 \LWR@htmltag{div class="marginblock" id="autofloat-\arabic{\LWR@thisfloat}"
20 style="width:\rndprintlength{\LWR@wrapwidth} ; %
21 \LWR@wrapposition"%
22 }%
23 \LWR@startpars
24 }
25
26
27 \NewDocumentEnvironment{wrapfigure}{o m o m}
28 {%
29 \LWR@subwrapfigure{#2}{#4}%
30 \captionsetup{type=figure}%
31 }
32 {%
33 \LWR@htmldivclassend{div}

```

```

34 }
35
36
37 \NewDocumentEnvironment{wraptable}{o m o m}
38 {%
39 \LWR@subwrapfigure{#2}{#4}%
40 \captionsetup{type=table}%
41 }
42 {%
43 \LWR@htmldivclassend{div}
44 }
45
46
47 \NewDocumentEnvironment{wrapfloat}{m o m o m}
48 {%
49 \LWR@subwrapfigure{#3}{#5}%
50 \captionsetup{type=#1}%
51 }
52 {%
53 \LWR@htmldivclassend{div}
54 }
55
56 \newlength{\wrapoverhang}

```

File 163 **lwarp-xcolor.sty**

§ 239 Package **xcolor**

Pkg xcolor xcolor is supported by lwarp.

\colorboxBlock and \fcolorboxBlock \colorboxBlock and \fcolorboxBlock are provided for increased HTML compatibility, and they are identical to \colorbox and \fcolorbox in print mode. In HTML mode they place their contents into a <div> instead of a . These <div>s are set to display: inline-block so adjacent \colorboxBlocks appear side-by-side in HTML, although text is placed before or after each.

Print-mode definitions for \colorboxBlock and \fcolorboxBlock are created by lwarp's core if xcolor is loaded.

background: none \fcolorbox and \fcolorboxBlock allow a background color of none, in which case only the frame is drawn, which can be useful for HTML.

color support Color definitions, models, and mixing are fully supported without any changes required.

tables Colored tables are ignored so far. Use CSS to style tables.

colored text and boxes `\textcolor`, `\colorbox`, and `\fcolorbox` are supported.

`\color` and `\pagecolor` `\color` and `\pagecolor` are ignored. Use CSS or `\textcolor` where possible.

§ 239.1 **Xcolor definitions: location and timing**

The lwarp core and its lwarp-xcolor package are tightly integrated to allow comparable results for print, HTML and print inside an HTML lateximage. This requires a number of definitions and redefinitions depending on whether each of xcolor and lateximage is being used, and whether print or HTML is being generated. Some of these actions are one-time when xcolor is loaded, and others are temporary as lateximage is used.

When xcolor is loaded in print mode: No special actions are taken at the time that xcolor is loaded in print mode, but see `\AtBeginDocument` below.

When lwarp-xcolor is loaded in HTML mode: xcolor's original definitions are saved for later restoration. `\LWR@restoreorigformatting` is appended to restore these definitions for use inside a lateximage. New HTML-mode definitions are created for `\textcolor`, `\pagecolor`, `\nopagecolor`, `\colorbox`, `\colorboxBlock`, `\fcolorbox`, `\fcolorboxBlock`, and `fcolorminipage`.

`\AtBeginDocument` in print or HTML mode: See Section 68.1. If xcolor has been loaded, the print-mode `\fcolorbox` is modified to accept a background color of none, and additional definitions are created for lwarp's new macros print-mode macros `\colorboxBlock`, `\fcolorboxBlock`, and `fcolorminipage`. The HTML versions of these macros will already have been created by lwarp-xcolor if it has been loaded.

For use inside an HTML lateximage, `\LWR@restoreorigformatting` is appended to temporarily set these functions to their print-mode versions.

In a lateximage in HTML mode: `\LWR@restoreorigformatting` temporarily restores the print-mode definitions of xcolor's functions. See `\LWR@restoreorigformatting` on page 330.

`\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 559.

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 559.

HTML lateximage: Remembers and reuses the print version.

\colorboxBlock:

Print: Becomes `\colorbox`.

HTML: Newly defined by lwarp-xcolor to use a `<div>`, page 560.

HTML lateximage: Remembers and reuses the print version `\colorbox`.

\fcolorbox:

Print: Modified to allow a background of none.

`\LWRprint@fcolorbox` at section 68.1

HTML: Redefined by lwarp-xcolor, page 560.

HTML lateximage: Remembers and reuses the print version.

\fcolorboxBlock:

Print: Becomes `\fcolorbox`. Section 68.1

HTML: Newly defined by lwarp-xcolor to use a `<div>`, page 561.

HTML lateximage: Remembers and reuses the print version `\fcolorbox`.

fcolorminipage:

Print: Newly defined in the lwarp core.

`LWRprint@fcolorminipage` at section 68.1

HTML: Newly defined by lwarp-xcolor, page 561.

HTML lateximage: Uses the print version.

\boxframe:

Print: Used as-is.

HTML: Redefined by lwarp-xcolor, page 562.

HTML lateximage: Remembers and reuses the print version.

§ 239.2 **Code**

for HTML output:

```
1 \LWR@ProvidesPackagePass{xcolor}
2 \begin{warpHTML}
```

Remember the following print-mode actions to be restored when inside a `lateximage` environment:

```
3 \LetLtxMacro\LWRprint@textcolor\textcolor
4 \LetLtxMacro\LWRprint@pagecolor\pagecolor
5 \LetLtxMacro\LWRprint@nopagecolor\nopagecolor
6 \LetLtxMacro\LWRprint@colorbox\colorbox
7 \LetLtxMacro\LWRprint@colorboxBlock\colorbox
8 \LetLtxMacro\LWRorigprint@fcolorbox\fcolorbox
9 \LetLtxMacro\LWRorigprint@fcolorboxBlock\fcolorbox
10 \LetLtxMacro\LWRorigprint@boxframe\boxframe
```

`\LWR@restoreorigformatting` Inside a `lateximage` the following gets restored to their print-mode actions:

```
11 \appto{\LWR@restoreorigformatting}{%
12 \LetLtxMacro\textcolor\LWRprint@textcolor%
13 \LetLtxMacro\pagecolor\LWRprint@pagecolor%
14 \LetLtxMacro\nopagecolor\LWRprint@nopagecolor%
15 \LetLtxMacro\colorbox\LWRprint@colorbox%
16 \LetLtxMacro\fcolorbox\LWRprint@fcolorbox%
17 \LetLtxMacro\boxframe\LWRorigprint@boxframe%
18 }
```

`\LWR@tempcolor` The color converted to HTML colorspace.

```
19 \newcommand*\LWR@tempcolor{}
20 \newcommand*\LWR@tempcolortwo{}
```

`\LWR@colorstyle` $\langle 1: styletext \rangle \langle 2: model \rangle \langle 3: color \rangle$

For a color style, prints the color converted to HTML colors.

```
21 \NewDocumentCommand{\LWR@colorstyle}{m m}{%
22 \begingroup%
23 \LWR@FBcancel%
```

Use the `xcolor` package to convert to an HTML color space:

```
24 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
```

Print the converted color:

```
25 \#\LWR@tempcolor%
26 \endgroup%
27 }
```

`\LWR@borderpadding` $\langle colorstyle \rangle$ $\langle color \rangle$ Prints the HTML attributes for a black border and padding. `\LWR@forceminwidth` must be used first in order to set the border width.

```
28 \newcommand*\LWR@borderpadding [2] {%
29 \uselengthunit{PT}%
30 border:\rndprintlength{\LWR@atleastonept} solid \LWR@colorstyle{#1}{#2} ; %
31 padding:\rndprintlength{\fboxsep}%
32 }
```

`\color` `\color` appears in the \LaTeX PDF output, but is ignored by `pdftotext` and thus is ignored in the HTML file. Text styling by local group is not yet supported.

Each of the following macros is given a temporary name, and is `\let` to the final name once the HTML conversion starts.

`\textcolor` $[\langle model \rangle]$ $\langle color \rangle$ $\langle text \rangle$

Converted into an HTML hex color span.

```
33 \RenewDocumentCommand{\textcolor}{O{named} m m}{%
34 \begingroup%
35 \LWR@FBcancel%
36 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
37 \InlineClass[color:\LWR@colorstyle{#1}{#2}]{textcolor}{%
38 \renewcommand*\LWR@currenttextcolor}{\#\LWR@tempcolor}%
39 #3%
40 }%
41 \endgroup%
42 }
```

`\pagecolor` $[\langle model \rangle]$ $\langle color \rangle$

Ignored. Use `\CSSFilename` instead.

```
43 \renewcommand*\pagecolor [2] [named] {}
```

`\nopagecolor` Ignored.

```
44 \let\nopagecolor\relax
```

`\colorbox` $[\langle model \rangle]$ $\langle color \rangle$ $\langle text \rangle$

Converted into an HTML hex background color ``.

```
45 \RenewDocumentCommand{\colorbox}{O{named} m +m}{%
```

```

46 \begingroup%
47 \LWR@FBcancel%
48 \uselengthunit{PT}%
49 \InlineClass[%
50 background:\LWR@colorstyle{#1}{#2} ; %
51 padding:\rndprintlength{\fboxsep}%
52 ]{colorbox}{#3}%
53 \endgroup%
54 }

```

`\colorboxBlock` [*model*] {*color*} {*text*}

Converted into an HTML hex background color <div>.

```

55 \NewDocumentCommand{\colorboxBlock}{0{named} m +m}{%
56 \begingroup%
57 \LWR@FBcancel%
58 \uselengthunit{PT}%
59 \begin{BlockClass}[%
60 background:\LWR@colorstyle{#1}{#2} ; %
61 padding:\rndprintlength{\fboxsep}%
62 ]{colorboxBlock}
63 #3
64 \end{BlockClass}%
65 \endgroup%
66 }

```

`\fcolorbox` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

67 \RenewDocumentCommand{\fcolorbox}{0{named} m 0{named} m +m}{%
68 \LWR@traceinfo{HTML fcolorbox #2 #4}%
69 \begingroup%
70 \LWR@FBcancel%
71 \uselengthunit{PT}%
72 \LWR@forceminwidth{\fboxrule}%
73 \ifthenelse{\equal{#4}{none}}%
74 {% no background color
75   \InlineClass[%
76     \LWR@borderpadding{#1}{#2}%
77   ]{fcolorbox}{#5}%
78 }%
79 {% yes background color
80   \InlineClass[%

```

```

81 \LWR@borderpadding{#1}{#2} ; %
82 background:\LWR@colorstyle{#3}{#4}%
83 ]{fcolorbox}{#5}%
84 }%
85 \endgroup%
86 }

```

`\fcolorboxBlock` [*framemodel*] {*framecolor*} [*boxmodel*] {*boxcolor*} {*text*}

Converted into a framed HTML hex background color span.

A background color of none creates a colored frame without a background color.

```

87 \NewDocumentCommand{\fcolorboxBlock}{0{named} m 0{named} m +m}{%
88 \LWR@traceinfo{HTML fcolorboxBlock #2 #4}%
89 \begingroup%
90 \LWR@FBcancel%
91 \uselengthunit{PT}%
92 \LWR@forceminwidth{fboxrule}%
93 \ifthenelse{\equal{#4}{none}}{%
94 {% no background color
95   \begin{BlockClass}[%
96     \LWR@borderpadding{#1}{#2}%
97   ]{fcolorboxBlock}
98   #5
99   \end{BlockClass}%
100 }%
101 {% yes background color
102   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
103   \begin{BlockClass}[%
104     background:\#\LWR@tempcolortwo; %
105     \LWR@borderpadding{#1}{#2}%
106   ]{fcolorboxBlock}
107   #5
108   \end{BlockClass}%
109 }%
110 \endgroup%
111 \LWR@traceinfo{HTML fcolorboxBlock done}%
112 }

```

Creates a framed HTML `<div>` around its contents.

A print-output version is defined in the lwarp core: section [68.1](#)

`\LWR@subfcolorminipage` {*framemodel*} {*framecolor*} {*background tag*} {*height*}

```

113 \NewDocumentCommand{\LWR@subfcolorminipage}{m m m m}{%
114 \begin{BlockClass}[%

```

```

115 #3%
116 \LWR@borderpadding{#1}{#2} ; %
117 \IfValueT{#4}{height:\rndprintlength{\LWR@tempheight} ; }%
118 width:\rndprintlength{\LWR@tempwidth}%
119 ]{fcolorminipage}%
120 }

```

Env fcolorminipage [*1:frame model*] [*2:frame color*] [*3:box model*] [*4:box color*] [*5:align*] [*6:height*] [*7:inner-align*] [*8:width*]

```

121 \NewDocumentEnvironment{fcolorminipage}{0{named} m 0{named} m 0{c} o o m}
122 {%
123 \LWR@FBcancel%
124 \setlength{\LWR@tempwidth}{#8}%
125 \IfValueT{#6}{\setlength{\LWR@tempheight}{#6}}%
126 \uselengthunit{PT}%
127 \LWR@forceminwidth{\fboxrule}%
128 \convertcolorspec{#1}{#2}{HTML}\LWR@tempcolor%
129 \ifthenelse{\equal{#4}{none}}%
130 {\LWR@subfcolorminipage{#1}{#2}{#6}}%
131 {%
132   \convertcolorspec{#3}{#4}{HTML}\LWR@tempcolortwo%
133   \LWR@subfcolorminipage{#1}{#2}{background:\#\LWR@tempcolortwo\ ; }{#6}%
134 }%
135 }
136 {\end{BlockClass}}

```

`\boxframe` [*width*] [*height*] [*depth*]

The depth is added to the height, but the box is not decended below by the depth.
`\textcolor` is honored.

```

137 \renewcommand*{\boxframe}[3]{%
138 {%
139 \setlength{\LWR@tempwidth}{#1}%
140 \setlength{\LWR@tempheight}{#2}%
141 \addtolength{\LWR@tempheight}{#3}%
142 \uselengthunit{PT}%
143 \LWR@forceminwidth{\fboxrule}%
144 \InlineClass[%
145 display:inline-block ; %
146 border:\rndprintlength{\LWR@atleastonept} solid \LWR@currenttextcolor{} ; %
147 width:\rndprintlength{\LWR@tempwidth} ; %
148 height:\rndprintlength{\LWR@tempheight}%
149 ]{\boxframe}{}%
150 }%
151 }

```

```
152 \end{warpHTML}
```

File 164 **lwarp-xfrac.sty**

§ 240 Package **xfrac**

Pkg xfrac Supported by adding xfrac instances.

for HTML output: 1 \LWR@ProvidesPackagePass{xfrac}

 **font size**

In the user's document preamble, lwarp should be loaded after font-related setup. During HTML conversion, this font is used by lwarp to generate its initial PDF output containing HTML tags, later to be converted by pdftotext to a plain text file. While the text may be in any font which pdftotext can read, the math is directly converted into SVG images using this same user-selected font. xfrac below is set for the Latin Modern (lmr) font. If another font is used, it may be desirable to redefine \xfracHTMLfontsize with a different em size.

\sfrac [*instance*] {<num>} [<sep>] {<denom>}

A text-mode instance for the default font is provided below. The numerator and denominator formats are adjusted to encase everything in HTML tags. \scalebox is made null inside the numerator and denominator, since the HTML tags should not be scaled, and we do not want to introduce additional HTML tags for scaling.

In math mode, which will appear inside a lateximage, no adjustments are necessary.

for HTML & PRINT: 2 \begin{warpall}

\xfracHTMLfontsize User-redefinable macro which controls the font size of the fraction.

```
3 \newcommand*\xfracHTMLfontsize{.6em}
```

```
4 \end{warpall}
```

for HTML output: 5 \begin{warpHTML}

font size A span for a small font, used in the numerator and denominator:

```
6 \newcommand*\LWR@htmlsmallfontstart{%
7 \LWR@htmltagc{span style="font-size:\xfracHTMLfontsize"{}}%
8 \LWR@nestspan%
9 %
10 }
11
```

```

12 \newcommand*{\LWR@htmlsmallfontend}{%
13 \LWR@htmltagc{/span}%
14 \endLWR@nestspan%
15 }

```

`\scalebox` A nullified `\scalebox` command, to avoid introducing HTML scaling tags:

```

16 \NewDocumentCommand{\LWR@noscalebox}{m o m}{#3}

```

instances Instances of `xfrac` for various font choices:

Produce HTML tags for a small superscript numerator and a small (non-subscript) denominator.

Scaling is turned off so that `pdftotext` correctly reads the result.

```

17 \DeclareInstance{xfrac}{default}{text}{
18 numerator-format = {%
19 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
20 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
21 denominator-format = {%
22 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
23 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

24 scaling = false
25 }
26
27 \DeclareInstance{xfrac}{lmr}{text}{
28 numerator-format = {%
29 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
30 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
31 denominator-format = {%
32 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
33 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},

```

For `pdftotext`, do not scale the text:

```

34 scaling = false
35 }
36
37 \DeclareInstance{xfrac}{lmss}{text}{
38 numerator-format = {%
39 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
40 \LWR@htmlsmallfontstart\textsuperscript{#1}\, \LWR@htmlsmallfontend},
41 denominator-format = {%
42 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
43 \LWR@htmlsmallfontstart{ }\, #1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```

44 scaling = false
45 }
46
47 \DeclareInstance{xfrac}{lmtt}{text}{
48 numerator-format = {%
49 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
50 \LWR@htmlsmallfontstart\textsuperscript{#1}\,\LWR@htmlsmallfontend},
51 denominator-format = {%
52 \LetLtxMacro{\scalebox}{\LWR@noscalebox}%
53 \LWR@htmlsmallfontstart{\,\#1\LWR@htmlsmallfontend},

```

For pdftotext, do not scale the text:

```

54 scaling = false
55 }

56 \end{warpHTML}

```

File 165 **lwarp-xltxtra.sty**

§ 241 Package **xltxtra**

Pkg xltxtra xltxtra is emulated.

(Based on original code by WILL ROBERTSON, JONATHAN KEW.)

for HTML output:

```

1 \LWR@ProvidesPackageDrop{xltxtra}

2 \RequirePackage{realscripts}
3 \RequirePackage{metalogo}
4 \newcommand*\TeX@logo@spacing[6]{}
5
6 \newcommand*\vfrac[2]{%
7 \textsuperscript{#1}/\textsubscript{#2}%
8 }
9
10 \newcommand\namedglyph[1]{%
11 \@tempcnta=\XeTeXglyphindex "#1"\relax
12 \ifnum\@tempcnta>0
13 \XeTeXglyph\@tempcnta
14 \else
15 \xxt@namedglyph@fallback{#1}%
16 \fi}
17

```

```

18 \newcommand\xxt@namedglyph@fallback[1]{[#1]}
19
20 \DeclareDocumentCommand{\showhyphens}{m}{}
```

File 166 **lwarp-xmpincl.sty**

§ 242 Package **xmpincl**

Pkg xmpincl Emulated.

for HTML output: Discard all options for lwarp-xmpincl:

```

1 \LWR@ProvidesPackageDrop{xmpincl}
2 \newcommand*{\includemp}[1]{}
```

File 167 **lwarp-xtab.sty**

§ 243 Package **xtab**

Pkg xtab xtab is emulated during HTML output.

for HTML output: 1 \LWR@ProvidesPackageDrop{xtab}

 misplaced alignment tab character &

For \tablefirsthead, etc., enclose them as follows:

```

\StartDefiningTabulars
\tablefirsthead
...
\EndDefiningTabulars
```

See section 7.7.

```

2 \newcommand{\LWRXT@firsthead}{}
3
4 \newcommand{\tablefirsthead}[1]{%
5   \long\gdef\LWRXT@firsthead{#1}%
6 }
7
8 \newcommand{\tablehead}[1]{
9
10 \newcommand{\tablelasthead}[1]{
11
12 \newcommand{\notablelasthead}{}

```

```
13
14 \newcommand{\tabletail}[1]{
15
16 \newcommand{\LWRXT@lasttail}{
17
18 \newcommand{\tablelasttail}[1]{%
19     \long\gdef\LWRXT@lasttail{#1}%
20 }
21
22 \newcommand{\tablecaption}[2][ ]{%
23     \long\gdef\LWRXT@caption{\caption[#1]{#2}}%
24 }
25
26 \let\topcaption\tablecaption
27 \let\bottomcaption\tablecaption
28
29 \global\let\LWRXT@caption\relax
30
31 \newcommand*{\shrinkheight}[1]{
32
33 \newcommand*{\xentrystretch}[1]{
34
35 \NewDocumentEnvironment{xtabular}{s o m}
36 {
37 \LWR@traceinfo{xtabular}
38 \table
39 \LWRXT@caption
40 \begin{tabular}{#3}
41 \TabularMacro\ifdefvoid{\LWRXT@firsthead}%
42 {\LWR@getmynexttoken}%
43 {\expandafter\LWR@getmynexttoken\LWRXT@firsthead}%
44 }
45 {%
46 \ifdefvoid{\LWRXT@lasttail}%
47 {}%
48 {%
49 \TabularMacro\ResumeTabular%
50 \LWRXT@lasttail%
51 }%
52 \end{tabular}
53 \endtable
54 \LWR@traceinfo{xtabular done}
55 }
56
57 \NewDocumentEnvironment{mpxtabular}{s o m}
58 {\minipage{\linewidth}\xtabular{#3}}
59 {\endxtabular\endminipage}
```

File 168 `lwarp-zwpageLayout.sty`

§ 244 Package **zwpageLayout**

Pkg zwpageLayout zwpageLayout is ignored.

```

for HTML output: 1 \LWR@ProvidesPackageDrop{zwpageLayout}

2 \def\noBboxes{}
3 \@onlypreamble\noBboxes
4
5 \expandafter\ifx\csname definecolor\endcsname\relax \else
6 \definecolor{cmykblack}{cmyk}{0,0,0,1}
7 \definecolor{grblack}{gray}{0}
8% \ifzwpl@redefineblack
9% \definecolor{black}{cmyk}{0,0,0,1}\color{black}
10% \fi
11 \definecolor{cmykred}{cmyk}{0,1,1,0}
12 \definecolor{cmykgreen}{cmyk}{1,0,1,0}
13 \definecolor{cmykblue}{cmyk}{1,1,0,0}
14 \definecolor{rgbred}{rgb}{1,0,0}
15 \definecolor{rgbgreen}{rgb}{0,1,0}
16 \definecolor{rgbbblue}{rgb}{0,0,1}
17% \ifzwpl@redefinetcmyk
18% \definecolor{red}{cmyk}{0,1,1,0}
19% \definecolor{green}{cmyk}{1,0,1,0}
20% \definecolor{blue}{cmyk}{1,1,0,0}
21% \fi
22 \fi
23
24 \let\OverprintXeTeXExtGState\relax
25
26 \DeclareRobustCommand\SetOverprint{\ignorespaces}
27 \DeclareRobustCommand\SetKnockout{\ignorespaces}
28 \DeclareRobustCommand\textoverprint[1]{\SetOverprint#1}
29 \DeclareRobustCommand\textknockout[1]{\SetKnockout#1}
30
31 \def\SetPDFminorversion#1{}
32 \@onlypreamble\SetPDFminorversion
33
34 \newcommand*\Vcorr{}
35
36 \DeclareRobustCommand\vb[1][ ]{}
37 \NewDocumentCommand{\NewOddPage}{* o}{}
38 \NewDocumentCommand{\NewEvenPage}{* o}{}

```

```
39 \def\SetOddPageMessage#\gdef\ZW@oddwarning}
40 \def\SetEvenPageMessage#\gdef\Z@@evenwarning}
41 \def\ZW@oddwarning{Empty page inserted}\let\ZW@evenwarning\ZW@oddwarning
42
43 \def\clap#1{#1}
44
45 \def\CropFlap{2in}
46 \def\CropSpine{1in}
47 \def\CropXSpine{1in}
48 \def\CropXtrim{.25in}
49 \def\CropYtrim{.25in}
50 \def\UserWidth{5in}
51 \def\UserLeftMargin{1in}
52 \def\UserRightMargin{1in}
53 \def\UserTopMargin{1in}
54 \def\UserBotMargin{1in}
55 \def\thePageNumber{#\,\arabic{page}}
56 \ifXeTeX
57 \def\ifcaseZWdriver{\ifcase2}
58 \else
59 \def\ifcaseZWdriver{\ifcase1}
60 \fi
61 \DeclareRobustCommand\ZWifdriver[2]{}
```

Change History and Index

§ 244 Change History

v0.10		Docs: Table: Float data structures. 310
General: 2016/03/08 Initial version . . .	1	Docs: Trademarks section. 384
v0.11		Docs: Troubleshooting
General: 2016/03/11	1	cross-references. 105
Added section: Operating-System		Test Suite: Assigned cleveref name
portability.	113	for Test Float. 1
Added section: Selecting the		Test Suite: Floatrow 1
operating system.	71	v0.15
Test Suite: limages and index in		General: 2016/04/06 1
README.txt	1	Added 426
Test Suite: MS-Windows in		Ampersand (&): Fixed handling
README.txt	1	when passed as an argument. . . 268
v0.12		Docs: Added warning icons for
\LWR@newhtmlfile: Bugfix: TOC		items needing special attention. 110
with numbered files.	225	Docs: Clarify print/HTML output. 72
General: 2016/03/14	1	Docs: Moved the supported
Global: Uses \p@(type) in float		functions table to the
captions.	1	introduction. 37
Test Suite: Sub-figures	1	Files: lwarp_formal.css added. . . . 1
v0.13		Fix: steps counter 426
\CaptionSeparator: Fix for newer		Fixed & handling. 424
babel package.	312	Test Suite: test_suite_formal.css
\LWR@LwarpStart: \up and \fup	241	file added. 1
General: 2016/03/24	1	v0.16
Removed package: subfig	1	General: 2016/04/11 1
Test Suite: Ordinals, Subcaption . . .	1	\titlingpage: Improved
tikzpicture: Fix dollar-redefined		print-output spacing. 248
bug for newer package.	529	Added XeLaTeX, LuaLaTeX
v0.14		support. 123
\LWR@htmlsectionfilename: Fix:		Docs: Font and UTF-8 support. . . 69
Links to home page.	195	Docs: Moved location of
General: 2016/03/31	1	\usepackage{lwarp}. 70
floatrow: Added.	421	Docs: Text not converting. 105
Docs: Commands for a successful		Fix: amsmath options clash 127
HTML conversion.	76	Fix: newtxmath compatibility. . . 127
Docs: Commands into a warpprint		Lwarp no longer selects fonts. 69, 122
environment.	73	Removed package: suffix 1
Docs: Newclude limitations.	97	Test Suite: Improved titlingpage. 248
Docs: Table: Cross-referencing		Test Suite: Lwarp no longer selects
data structures.	300	fonts. 1

Test Suite: Supports XeLaTeX, LuaLaTeX.	1	verse: Supports verse, memoir packages.	552
xfrac: Adjusted for the use of any font:	564	minipage: Fix: \linewidth, \textwidth, \textheight inside a minipage.	361
v0.17		v0.19	
\LWR@htmlsectionfilename: Fix: Links when entire doc is one HTML page.	195	\HTMLFilename: Docs: Escape filename underscores.	194
General: 2016/04/14	1	\HomeHTMLFilename: Docs: Escape filename underscores.	194
mdframed: Added.	464	\LWR@LwarpStart: Enabled \ equal to \newline.	239
Test Suite: Fix: Print-version front-matter page numbers.	1	\LWR@doubledollar: MathJax support.	328
Test Suite: Mdfamed	1	\LWR@filestart: lwarp_mathjax.txt loaded.	237
v0.18		\LWR@hspace: Fix: \hspace length computations.	376
\LWR@hspace: \hspace supported.	376	\LWR@minipagestartpars: Surpresses paragraph tags between minipages.	374
\LWR@includegraphicsb: Add: svgz file extension.	441	\LWR@singledollar: MathJax support.	328
em, ex, %, px dimensions preserved.	441	\LWR@tabledatamultirowtag: Added optional args.	478
Fix: \linewidth, \textwidth, \textheight inside a minipage.	441	\LateximageFontSizeName: Add: User-adjustable math/lateximage font size.	344
Improved HTML output linebreaks.	441	\minipagefullwidth: Added: No width tag for the next minipage in HTML.	361
\LWR@myshorttoc: Reorganize \HomeHTMLFilename logic.	316	\rule: Added	380
\LWR@newhtmlfile: sideroc after title, improving responsive design.	224	\warpHTMLonly: Added.	117
\LWR@requesttoc: Reorganize \HomeHTMLFilename logic.	242	\warpprintonly: Replaces \rowprintedonly.	117
\LWR@subhyperref: Improved HTML output linebreaks.	308	\xfracHTMLfontsize: Added.	563
\LWR@subhyperrefclass: Improved HTML output linebreaks.	308	General: 2016/06/08	1
\LWR@subinlineimage: Surpress extra space.	309	Avoids MathJax.	327
General: 2016/05/19	1	cleveref: Loaded \AtEndPreamble.	357
File: lwarp.css: Improved TOC outline display.	1	CSS for table note item.	528
Files: lwarp.css and lwarp_formal.css: Improved responsive design.	1	Docs: multirow browser bug workaround.	93, 264
Microtype disabled during HTML generation	123	Docs: Math options.	70
PDF Unicode input characters.	111	Docs: Table: Cross-referencing data structures, updated.	300
Test Suite: Verse package	1	File: lwarp_mathjax.txt added.	1
lateximage: pdfcrop: --hires added.	346	File: lwarp.css: tnoteitemheader added	1
Reorganize \HomeHTMLFilename logic.	346		
Surpress extra space.	346		

Introduction: MathJax support mentioned.	36	\LWR@section: Combined higher-level sections together into files.	230
MathJax support added. 329, 335, 336		\LWR@setOSWindows: Auto-detects operating system.	113
Options: mathsvg and mathjax	115	\LWR@subhtmlclass: Factored code.	205
Supports colored \rule.	558	\SetHTMLFileNumber: Add: Control file numbers.	194
titles: null \pagestyle and \thispagestyle for HTML.	529	\cpagerefFor: User-redefinable word for page references.	357
tikzpicture: Adapts to tikz version.	529	\dotfill: Inserts an ellipsis.	373
equation: MathJax support.	334	\hfill: Inserts a \qqquad.	373
v0.20		\hrulefill: Inserts a short rule.	373
\BlockClassSingle: Renamed from "LWR@htmldivclassline".	207	\hyperindexref: Print mode provided in case hyperref not used.	324
\HTMLDescription: Added \NewHTMLdescription. (Renamed in v0.30.)	217	\pageref: Added.	307
\HTMLFilename: No longer escape underscores.	194	\tracinglwrap: Added.	132
\HomeHTMLFilename: No longer escape underscores.	194	General: 2017/02/09	1
\InlineClass: Renamed from "inlineclass".	207	afterpage: Added.	389
\LWR@LwrapStart: Fix: math cross references.	241	alltt: Added.	390
\LWR@closeparagraph: \unskip extra spaces.	210	bookmark: Added.	397
No break tags in the start/end of a tabular.	210	caption and subcaption supported. 1	
\LWR@endofline: Fix: \ \	374	cleveref and referencing patches: Applied \AfterEndPreamble.	357
\LWR@filestart: Adds meta description.	237	draftwatermark: Added.	403
\LWR@hspace: Add: Supports HTML thin breakable space.	376	eso-pic: Added.	407
\LWR@htmldivclass: Added optional style.	205	everypage: Added.	407
\LWR@htmlclass: Added optional style.	205	extramarks: Added.	408
\LWR@htmlsectionfilename: HTMLFilename: removed additional trailing '-', and may be empty.	195	fancyhdr: Added.	412
Sections called "Index" or "index" have an underscore prepended to their filenames if no prefix.	195	hyperref: Additional user macros.	447
\LWR@includegraphicsb: Fix: \linewidth in a floatrow.	441	keyfloat: Added.	453
Fix: Expands filename.	441	letterspace: User-interface emulated.	455
\LWR@longtabledatacaptiontag: Fix: Pars in captions.	288	listings: Added.	457
		lrcaption: Added.	462
		lwrap-newproject: Added.	137
		microtype: User-interface emulated.	473
		needspace: Added.	480
		nowidow: Added.	482
		placeins: Added.	495
		ragged2e: Added.	497
		setspace: Improved support.	503
		textpos: Added.	523
		titles: Added.	529
		titlesec: Added.	532
		titletoc: Added.	534
		titling: Improved compatibility.	535

tocloft: Added.	542	newunicodechar: Added.	481
wallpaper: Added.	553	lwarpmk: Fix: lwarpmk again for Windows.	178
wrapfig: Added.	554	lwarpmk: Fix: lwarpmk images for Windows.	178
Added @, <, > columns.	265	lwarpmk: Fix: lwarpmk uses lateximages text file instead of shell script.	178
Added single-expansion data arrays.	193	Add: Errors for misplaced packages.	118
Code factored into independent lwrap_html files.	384	Docs: Added internet class. . . .	41
Docs: Examples for generating HTML file names.	62	Docs: Added TeX2page, GladTeX. . .	41
Docs: Improved index.	1	Docs: Installing on Windows. . . .	47
Enhanced titling support.	247	File lwrap_tutorial.txt added. . . .	51
File: lwrap.css: Minor fixes for validation.	1		
File: lwarpmk used to compile print, HTML, indexes, and lateximages.	1	v0.22	
Fix: \linewidth in a floatrow. . .	424	\LWR@parseDcolumn: Added tabular D column.	275
Improved float caption type handling.	419	\LWR@parsebangcolumn: Added tabular ! column.	272
Moved sidebar and example code to test suite.	1	\LWR@parsetablecols: Unknown table column types become l. Added tabular D, !, X columns. . . .	277
Page geometry set to 6in wide with large margins.	124	\LWR@printmccoldata: Added tabular D, !, and X columns. . . .	284
Parallel versions of aux files for print/HTML.	1	General: 2017/03/02	1
Removed reliance on make, grep, gawk.	1	abstract: Added.	386
Tabular: \unskip extra spaces. . .	265	changepage: Added.	399
Test Suite: HTML meta descriptions.	1	dcolumn: Added.	402
Verbatim: Added.	414	ftnright: Added.	434
verbatim: Added.	258	geometry: Nullified commands. . .	435
BlockClass: Added optional style. .	207	indentfirst: Added.	453
Renamed from "blockclass". . . .	207	layout: Added.	454
LWR@nestspan: Fix: Minipages inside a span.	202	lscap: Added.	462
		mcaption: Added.	464
v0.21		nameref: Added.	480
\LWR@LwrapStart: Changed lateximages to a .txt file. . . .	239	nextpage: Added.	481
\LWR@filestart: Skip title if not given.	237	parskip: Added.	494
\LWR@newhtmlfile: Skip title if not given.	224	showkeys: Added.	505
\marginpar: Fixed source listing. . .	221	sidecap: Added.	505
General: 2017/02/23	1	tabularx: Added.	518
fontenc: Added.	428	varioref: Supported.	82
fontspec: Added.	429	verse: Added.	551
inputenc: Added.	453		
newclude: Added.	480	v0.23	
		\LWR@parsetablecols: Fix for vert bar column type.	277
		\LWR@printmccoldata: Fix for vert bar column type.	284
		General: 2017/03/02	1

v0.24		footnote: Added.	430
	\LWR@hspace: Add: \hspace \fill converts to 2em	marginfix: Added.	463
		marginnote: Added.	463
	\LWR@htmlfileref: Fix: Index links while \tracinglwrap.	mparhack: Added.	476
		pagenote: Supported as-is.	493
	\hypertocfloat: List of floats responds to lofdepth, lotdepth.	sidenotes: Added.	506
		lwrap.css: Improved responsive marginpar and marginblock.	139
	General: 2017/03/15	Docs: Improved MiKTeX install instructions.	44, 47
		Dollar span avoided in a lateximage.	327
	floatrow: Support for subfig.	Footnotes now are \LaTeX boxes instead of pagenotes.	217
	subfig: Added.	lateximage: Labels track page numbers of lateximages.	346
	tikz: For tikz v3.0.0 or later, auto-loads tikz babel library if necessary.	Print mode now uses a minipage of \linewidth.	346
		picture: Fix for \makebox in picture.	358
	Docs: Filename underscore.		
	No longer preloads subcaption; conflicted with subfig.		
	picture: Fix for inline images.		
	tikzpicture: Fix for inline images.		
v0.25			
	\LWR@loadnever: Added the ability to prevent conflicting packages.		
		v0.27	
	\addcontentsline: Handles theorems.	\LWR@footnotetext: Fix for table footnote par tags.	218
		General: 2017/04/04	1
	General: 2016/03/22	lettrine: Added.	455
		microtype: Fix with Xe \LaTeX , Lua \LaTeX	473
	amsthm: Added.	soul: Added.	508
	ccaption: Prevented.	ulem: Added.	550
	ellipsis: Added.	Docs: Installing utilities for MacOS.	49
	emptypage: Added.	Docs: Limitations of saveboxes.	78
	framed: Added.	Page geometry modified to reduce line overflow.	124
	lips: Added.		
	mdframed: Help avoid hyphenation.	v0.28	
		\@wrindex: Improved indexing.	324
	ntheorem: Added.	\HTMLAuthor: Added \HTMLauthor. (Renamed in v0.30.)	216
	showidx: Added.	\LWR@LwrapEnd: If FormatEpub or FormatWordProcessor, no bottom nav.	243
	theorem: Added.	\LWR@LwrapStart: FormatWordProcessor forces single-file output.	239
	Basic \LaTeX theorems: improved css.	\LWR@filestart: Adds HTML meta author.	237
		\LWR@forcenewpage: Forces new PDF page before major environments.	198
	Docs: Adds credits for patched code.		
	Docs: Testing lwrap.		
	Fix: Allows Xe \LaTeX and Lua \LaTeX to preload graphics and graphicx.		
v0.26			
	General: 2017/03/31		
	cutwin: Added.		
	endnotes: Added.		
	floatflt: Added.		
	footmisc: Added.		
	footnotehyper: Added.		

\LWR@htmlcomment: Breaks ligatures in HTML comments.	204	\HTMLAuthor: Renamed from \HTMLauthor.	216
\LWR@includegraphicsb: Adapts to graphics syntax.	441	\HTMLDescription: Renamed from \NewHTMLdescription.	217
\LWR@newhtmlfile: If FormatEPUB or FormatWordProcessor: skips headers, footers, nav.	224	\HTMLFirstPageTop: Renamed from \SetFirstPageTop.	215
\LWR@parsetablecols: Added L, C, R, J column types.	277	\HTMLLanguage: Renamed from \MetaLanguage.	237
\LWR@startref: Removed space. . .	305	\HTMLPageBottom: Renamed from \SetPageBottom.	215
\chapter: If EPUB, prints footnotes before each section.	235	\HTMLPageTop: Renamed from \SetPageTop.	215
\hyperindexref: Improved indexing.	324	General: 2017/04/29	1
\textup: Fixed span class.	369	lwrap-newproject removed, and combined with lwrap.	137
General: 2017/04/14	1	lwrapmk: Add: xdyfile configuration option.	178
glossaries: Added.	436	lwrapmk: Fix: xindy and texindy adjusted for pdflatex, xelatex and lualatex.	178
graphics: Added.	437	lwrapmk: Fix: xindy now used for print index generation with latexmk.	178
tabularx: Fix for optional pos. . .	518	lwrapmk: language now used for both index and glossary generation.	178
tabulary: Added.	519	File: lwrap_html.xdy renamed to lwrap.xdy.	175
lwrapmk: Add: printglossary and htmlglossary commands.	178	Fix: *.css files only written in print mode.	139
Added boolean FormatEPUB.	133	Fix: lwrap.xdy only written in print mode.	175
Added boolean FormatWordProcessor.	133	Fix: lwrap_mathjax.txt only written in print mode.	176
Added boolean HTMLDebugComments.	132	Option OSWindows replaces macro \warpOSWindows.	116
Added boolean HTMLMarkFloats	133	Option latexmk replaces macro \UseLatexmk.	116
Docs: Modfying lwrapmk and index processing.	103	Option lwrapmklang changed to IndexLanguage.	115
File lwrap_mathjax.txt: Updated CDN repository.	176	Option xdyFilename added.	115
Forced onside to maintain large right margin.	124	Options HomeHTMLFilename and HTMLFilename replace macros \HomeHTMLFilename and \HTMLFilename.	116
v0.29		v0.31	
\LWR@includegraphicsb: Fix: Error when no optional arguments.	441	General: 2017/05/15	1
General: 2017/04/15	1	keyfloat: Improved compatibility.	453
lwrapmk: Add: language option for config files.	178		
Add: lwrapmklang option for lwrap.	115		
Docs: Using a glossary	67		
File *.lwrapmkconf: Add: language option for config files.	138		
File lwrapmk.conf: Add: language option for config files.	138		
v0.30			
\CSSFilename: Renamed from \NewCSS.	216		

v0.32		\LWR@html element classline:	
	\RequirePackage: Fix: Ignores blanks in package list.	Moved optional argument in front of mandatory.	206
	General: 2016/06/09	\LWR@html span class: Moved optional argument in front of mandatory.	203
	glossaries: Prevent error with \glo@name not defined.	\LWR@nullfonts: Improved font control.	370
	lwarpmk: Fix: io.lines() changed to file:lines() due to luatex changes.	\LWR@restoreorigformatting: booktabs: Works inside lateximage.	330
v0.33		Improved font control.	330
	\HTMLAuthor: Fix: Provides empty default author if none given. . .	\LWR@subhtml element class: Moved optional argument in front of mandatory.	205
	\LWR@loadbefore: Fix: No \PackageError if already loaded.	\LWR@tabledata column tag: booktabs: Works inside lateximage.	294
	\LWR@parseat column : Fix: Column alignment with leftmost @. . . .	\fboxBlock: Added.	365
	\LWR@tabledata single column tag: Fix: Macros in tabular could cause extra data cell.	\makebox: Fix: Handles paren arg. .	363
	\LWR@vspace: Add: \vspace nullified.	General: 2017/08/08	1
	\StartDefiningTabulars: Add: Avoids error: misplaced alignment tab character &.	babel-french: Adds fixed-width HTML spaces to punctuation. . .	200
	General: 2017/07/10	balance: Added.	395
	amsmath: Removed fleqn option. . .	booktabs: Works inside lateximage.	296, 398
	fancyhdr: Fix: Optional args for \thead, etc.	boxedminipage2e: Added.	398
	Add: Tabular at and bang columns now have their own HTML columns.	boxedminipage: Prevented.	398
	cleveref: Fix: Loaded \AtEndPreamble.	crop: Added.	400
	Fix: Incorrectly-inline math environments.	enumerate: Added.	405
	New handling of & to localize catcode changes.	enumitem: Added, no longer required.	405
v0.34		everyshi: Added.	408
	\@fnsymbol: Text symbols instead of math.	fancybox: Added.	410
	\InlineClass: Moved optional argument in front of mandatory. .	fancyvrb: Added, no longer required.	413
	\LWR@html div class: Moved optional argument in front of mandatory.	figcaps: Added.	418
	\LWR@html element class: Moved optional argument in front of mandatory.	filecontents: Required. Patched for morewrites.	125
		floatpag: Added.	421
		flushend: Added.	427
		fullpage: Added.	434
		hyperxmp: Added.	452
		idxlayout: Added.	452
		marginfit: Added.	463
		mdframed: Improved mdtheorem patch.	470
		moreverb: Added.	474
		paralist: Added.	493
		pdflscape: Added.	494

pdfsync: Added.	494	Force HTML superscripts.	218
prelim2e: Added.	495	\LWR@nullfonts: Fix: Filenames	
rotfloat: Added.	501	while using MathJax.	370
savetrees: Added.	502	\LWR@restoreorigformatting:	
shadow: Added.	504	siunitx: Improved	
syntonly: Added.	517	super/subscripts in a	
titles: No longer required.	529	lateximage.	330
titleref: Prevented.	532	\LWR@section: Improved spacing.	230
xmpincl: Added.	566	\LWR@stoppars: Extra HTML source	
Added.	561	space after paragraphs.	213
Docs: Horizontal space		\fbox: Fix: Uses \fboxrule and	
limitations.	1	\fboxsep.	365
Docs: Misplaced alignment		\framebox: Fix: Handles width and	
character.	105	horiz position.	364
File: README.txt: updated.	1	\makebox: Fix: Handles width and	
File: lwrap_mathjax.txt version		horiz position.	363
change.	176	General: 2017/08/17	1
Fix: Added the eqnarray		babel-french: Adjustements for	
environments.	336	French variants, load order,	
Improved font control.	368	footnotes, ellipses.	200
Lists refactored to remove		footnote: Extra HTML source space	
enumitem requirement.	259	after paragraphs.	430
Verbatim refactored to remove		siunitx: Fix for babel-french.	351
fancyvrb requirement.	256	siunitx: Improved symbol support.	508
lateximage: Fix: lateximage with		transparent: Added.	547
minipage, \parbox, \makebox,		upref: Added.	551
\fbox, \framebox, \raisebox,		xcolor: Added \fcolorboxBlock,	
\scalebox, \reflectbox.	346	\colorboxBlock.	556
BlockClass: Moved optional		xcolor: Fix: Background none in	
argument in front of mandatory.	207	print mode.	556
fminipage: Added.	366	xcolor: Refactored	
\LWR@nestspan: Fix: Minipages,		\LWR@colorstyle.	558
BlocksClass, and lists inside a		xcolor: Uses \fboxrule and	
span.	202	\fboxsep.	556
\LWR@tabular: booktabs: Works		xcolor: \fcolorbox etc. now work	
inside lateximage.	297	inside lateximage.	556
v0.35		Docs: Reorganized: Special cases	
General: 2017/08/08	1	and limitations.	76
Fix: \textbf and related.	368	Source: Improved formatting.	1
v0.36		lateximage: Footnotes appear in	
\LWR@HTMLsanitize: Fix for		regular text instead of the	
babel-french.	345	lateximage minipage.	346
\LWR@HTMLsanitizeexpand: Fix for		\LWR@tabular: Fix for babel-french.	297
babel-french.	345	v0.37	
\LWR@closeparagraph: Extra HTML		\include: Maintains independent	
source space after paragraphs.	210	.aux files for HTML.	130
\LWR@currenttextcolor: Fix for		General: 2017/08/19	1
\rule when xcolor not loaded.	378	L ^A T _E X accents: Added.	135
\LWR@footnotetext: Extra HTML		babel-french: Adjustment for load	
source space after paragraphs.	218	order.	200

color: Prevented.	400	v0.39	\@maketitle: titling version.	538
comment: Maintains independent			Native \LaTeX version.	251
cutfiles for print, HTML.	117		Removed minipages.	251, 538
siunitx: Improved symbol support. . . .	508		Supports authblk with <div>s of	
textcomp: Improved support.	520		class oneauthor instead of	
lwarpmk: Removes additional			tabular.	251, 538
HTML aux files.	178		\AddSubtitlePublished: Added.	252
File handles reorganized.	130		\LWR@domulticolumn: Add: Optional	
v0.38			vpos and # rows.	286
\@secntformat: Added for			\LWR@restoreorigformatting:	
appendix.	229		Appended with \appto instead of	
\ForceHTMLPage: Added.	227		calling various macros.	330
\ForceHTMLTOC: Added.	227		\LWR@tabledatacolumnntag: Don't	
\LWR@section: \part* starts a new			start a data cell if see	
HTML page, for appendix.	230		\TabularMacro.	294
Modified spacing, uses			\LWR@tabledatamultirowtag: Add:	
\numberline.	230		New optional vpos argument.	478
\numberline: Added trailing \quad. . . .	320		Add: Supports left/right border for	
\part: Fix with article class.	235		bigdelim.	478
General: 2017/08/27	1		Fix: Long text argument.	478
appendix: Added.	394		\ResumeTabular: Added.	293
arabicfront: Added.	394		\TabularMacro: Added.	293
caption2: Prevented.	399		\multicolumnrow: Added.	291, 479
chappg: Added.	400		\printauthor: Removed minipages. . . .	247
color: Forces xcolor as well.	400		Supports authblk with <div>s of	
fix2col: Added.	419		class oneauthor instead of	
fncychap: Added.	427		tabular.	247
grffile: Added.	447		\thanksmarkseries: Removed	
metalogo: Added.	473		minipage footnotes.	539
nonumonpart: Added.	481		General: 2017/09/05	1
nopageno: Added.	482		a4wide: Added.	385
pagenote: Option page disabled.	493		a4: Added.	385
realscripts: Added.	498		a5comb: Added.	385
relsize: Added.	499		addlines: Added.	388
romanbarpagenumber: Added.	500		anysize: Added.	393
romanbar: Added.	500		authblk: Added.	394
scalefnt: Added.	502		bigdelim: Added.	396
siunitx: Removed from lwarp core. . . .	508		bigstrut: Added.	397
textcomp: Removed from lwarp			chngepage: Prevented.	400
core.	520		ebook: Added.	403
tocbibind: Added.	541		fullwidth: Added.	435
xltxtra: Added.	565		supertabular: Added.	516
lwarpmk: Added print1 and html1			textarea: Added.	519
actions.	178		titling: Improved compatibility.	535
Added \markboth, \sloppy, etc.	198		titling: Removed extraneous	
Docs: Enhanced <i>Supported</i>			center environments.	536
<i>Functions</i> table.	37		typearea: Added.	549
Docs: Index, tocbibind.	86		xtabular: Added.	566
Docs: Starred sections.	83		zwpagelayout: Added.	568

Docs: Reorganized tabular discussion.	92	<code>\simplechapterdelim</code> : Added for <code>tocbibind</code> , <code>anonchap</code>	230
Titlepage <code>\published</code> and <code>\subtitle</code> removed. <code>\AddSubtitlePublished</code> restores.	252	<code>\underline</code> : Added.	372
<code>titlepage</code> : Clear pending footnotes. Removed minipages.	246	General: 2017/09/25	1
<code>titlingpage</code> : Clear pending footnotes.	536	<code>adjmulticol</code> : Added.	387
v0.40		<code>anonchap</code> : Added.	393
<code>\@chapcntformat</code> : Added for <code>tocbibind</code> , <code>anonchap</code>	230	<code>bigdelim</code> : Improved documentation.	396
<code>\LWR@hline</code> : Added.	297	<code>cuted</code> : Added.	401
<code>\LWR@includegraphicsb</code> : Add: Full <code>\graphicspath</code> support.	441	<code>dblfnote</code> : Added.	402
<code>\LWR@nullfonts</code> : Fix: Long arguments for expandable command.	370	<code>fnpos</code> : Added.	428
<code>\LWR@restoreorigformatting</code> : Improved \LaTeX logos inside a <code>lateximage</code>	330	<code>graphics</code> : Moved out of the <code>lwarp</code> core.	437
Improved symbols inside a <code>lateximage</code>	330	<code>graphics</code> : Restores <code>\includegraphics</code> and <code>\DeclareGraphicsExtensions</code> in a <code>lateximage</code>	437
Nullified <code>\InlineClass</code> , etc. inside a <code>lateximage</code>	330	<code>graphicx</code> : Moved out of the <code>lwarp</code> core.	447
<code>\LWR@tabledatacolumnstag</code> : Fix for <code>bigdelim</code> : <code>\ldelim</code> , <code>\rdelim</code>	294	<code>grffile</code> : Directly supported.	447
<code>\LWR@tabledatamultirowtag</code> : Improved <code>bigdelim</code> borders.	478	<code>midfloat</code> : Added.	474
<code>\chapter</code> : Added support for <code>quotchap</code>	235	<code>pnote</code> : Added.	495
<code>\multicolumnrow</code> : Fix: Adapts to older <code>multirow</code> and <code>xparse</code>	291	<code>quotchap</code> : Added.	496
		<code>sectsty</code> : Added.	502
		<code>stabular</code> : Added.	510
		<code>tabls</code> : Added.	518
		<code>textcomp</code> : Additional symbols, improved XeLaTeX and LuaLaTeX support.	520
		<code>tocbibind</code> : Improved for <code>\simplechapter</code>	541
		<code>xltxtra</code> : Fix for <code>\showhyphens</code> with XeLaTeX.	565
		No longer preloads <code>xfrac</code>	127

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	
$\$$	<i>327</i>
$\&$	<i>194, 4839</i>
\langle	<u>6019</u>
\rangle	<u>6019</u>
$\,$	<i>76</i>
$\@setcpageref$	<u>6743</u>
$\@setcref$	<u>6739</u>
$\@setcrefrange$	<u>6740</u>
$\@author$	<i>245</i>
$\@begintheorem$	<u>4685</u>
$\@capttype$	<u>5708</u>
$\@chapcntformat$	<u>4062</u>
$\@currentlabelname$	<u>5488</u>
$\@date$	<i>245</i>
$\@dlbfloat$	<u>5686</u>
$\@endtheorem$	<u>4695</u>
$\@ensuredmath$	<u>6026</u>
$\@float$	<u>5686</u>
$\@fnsymbol$	<u>4507</u>
$\@footnotetext$	<u>3831</u>
$\@include$	<u>450</u>
$\@makecaption$	<u>5710</u>
$\@makefnmark$	<u>3803</u>
$\@makefntext$	<u>3802</u>
$\@maketitle$	<i>49, 4530</i>
$\@mpfootnotetext$	<u>3832</u>
$\@opargbegintheorem$	<u>4690</u>
$\@seccntformat$	<u>4060</u>
$\@title$	<i>245</i>
$\@wrglossary$	<u>5960</u>
$\@wrindex$	<u>5953</u>
\backslash	<i>374</i>
$\$$	<i>328</i>
$\$\$$	<i>328</i>
\sim	<i>76</i>
A	
a4 (package)	<i>385</i>
a4wide (package)	<i>385</i>
a5comb (package)	<i>385</i>
abstract (environment)	<u>4593</u>
abstract (package)	<i>385</i>
\abstractname	<i>62, 4592</i>
accents	<i>227</i>
\addcontentsline	<u>5762</u>
addlines (package)	<i>388</i>
\AddSubtitlePublished	<u>4555</u>
adjmulticol (package)	<i>387</i>
Adobe (program)	<i>43</i>
\affiliation	<u>4417</u>
afterpackage (package)	<i>124</i>
afterpage (package)	<i>389</i>
algorithmic with newfloat, trivfloat	<i>549</i>
algorithmicx (package)	<i>389</i>
align (environment)	<u>6353</u>
align* (environment)	<u>6376</u>
alignment tab character &, misplaced	<i>92, 263, 268</i>
alltt (package)	<i>390</i>
\AmS	<u>7262</u>
AMSMath split miss-numbered .	<i>87, 89, 325, 483</i>
amsmath (package)	<i>127</i>
amsthm (package)	<i>390</i>
\and	<i>244</i>
anonchap (package)	<i>393</i>
anysize (package)	<i>393</i>
appendix (package)	<i>393</i>
arabicfront (package)	<i>394</i>
array (package)	<i>300</i>
AsciiDoc (program)	<i>42</i>
AsciiDoctor (program)	<i>42</i>
Asciidoctor-LaTeX (program)	<i>42</i>
\attrib	<i>96, 255, 552</i>
\attribution	<u>4603</u>
authblk (package)	<i>394</i>
author HTML meta tag	<i>74, 216</i>
\author	<i>74, 244</i>
autosec	<i>228</i>
B	
babel (package)	<i>97</i>
\backmatter	<u>4044</u>

geometry (package)	124, 435	\HTMLPageBottom	63, 245, 3766
getttitlestring (package)	125	\HTMLPageTop	63, 3758
GladTeX (program)	42	\HTMLUnicode	3239
glossaries (package)	85, 435	HTMLvleftskip (length)	96, 255, 552
glossary		\hyperindexref	5967
language	67	hyperref (package)	307, 447
processing	67	\hypertoc	5877
glyphtounicode (file)	70	\hypertocfloat	5890
graphics (package)	351, 437	hyperxmp (package)	452
graphicx (package)	351, 447		
grffile (package)	447		
		I	
		icon	
		warning	110
		idxlayout (package)	452
		ifplatform (package)	112
		images	
		graphicx package	351
		in strange places	106
		showing as HTML	57, 106
		\include	450
		\includegraphics	181
		indentfirst (package)	453
		InDesign (program)	43
		index	
		language	67
		placement and toc options	86
		processing	55, 56
		UTF-8	70
		IndexLanguage (option) 67, 86, 115, 436, 540	
		\InlineClass	3597
		inputenc (package)	69, 453
		internet (class)	41
		item	
		empty	77, 259
		itemize (environment)	4707
		\itshape	7063
		J	
		JavaScript	
		MathJax	87, 325
		K	
		\kern	76
		keyfloat (package)	453
		kvoptions (package)	114
		L	
		label	
		in HTML	241
		math environment	335
H			
Hevea (program)	41		
\hfill	7073		
\HomeHTMLFilename	3250		
HomeHTMLFilename (option)	60, 116		
horizontal space	76		
between minipages	374		
\href	5631		
\hrulefill	7074		
\hskip	76		
\hspace	76, 375, 7170		
HTML			
commands for a successful conver-			
sion	76		
conversion settings	60		
debug comments	132		
EPUB	98, 133		
word processor	100, 101, 133		
filename generation	62		
headings	108		
in place of SVG images	57, 106		
meta tag			
author	74, 216		
description	74, 217		
generator	238		
viewport	238		
tabular column conversion	276		
\HTMLAuthor	63, 74, 3787		
HTMLDebugComments (boolean)	132		
\HTMLDescription	63, 74, 3793		
\HTMLentity	3231		
\HTMLFilename	3249		
HTMLFilename (option)	60, 116		
\HTMLFirstPageTop	63, 3750		
htmlglossary (option) [lwarpmk]	86, 436		
\HTMLLanguage	63, 4268		
HTMLleftmargini (length) 96, 255, 256, 552			
HTMLMarkFloats (boolean)	101, 133		

language	<u>7248</u>
glossary	<u>67</u>
index	<u>67</u>
language HTML metadata	<u>237</u>
\LaTeX	<u>7236</u>
LaTeX2HTML (program)	<u>41</u>
\LaTeXe	<u>7236</u>
lateximage (environment)	<u>6498</u>
\LateximageFontSizeName	<u>6456</u>
lateximages	
font size	<u>87, 325, 344</u>
latexmk (option)	<u>60, 116</u>
LaTeXML (program)	<u>41</u>
layout (package)	<u>454</u>
lengths:	
HTMLleftmargini	<u>96, 255, 256, 552</u>
HTMLvleftskip	<u>96, 255, 552</u>
LWR@minipageheight	<u>359</u>
LWR@minipagewidth	<u>359</u>
LWR@tempheight	<u>376</u>
LWR@tempraise	<u>376</u>
LWR@tempwidth	<u>376</u>
\VerbatimHTMLwidth	<u>256</u>
vleftmargini	<u>96, 255, 552</u>
vleftskip	<u>96, 255, 552</u>
letltxmacro (package)	<u>112</u>
letterspace (package)	<u>455</u>
lettrine (package)	<u>455</u>
LibreOffice (program)	<u>42</u>
ligatures	<u>69, 70, 123</u>
line numbers	<u>110</u>
\linebreak	<u>7173</u>
\LinkHome	<u>3300</u>
Linux (program)	<u>71, 113</u>
lips (package)	<u>456</u>
list item, empty	<u>77, 259</u>
listings (package)	<u>457</u>
\listof	<u>5843</u>
lmodern (package)	<u>69</u>
lofdepth (counter)	<u>321</u>
longtable (environment)	<u>2</u>
longtable (package)	<u>460</u>
lotdepth (counter)	<u>321</u>
lscope (package)	<u>462</u>
ltxcaption (package)	<u>462</u>
LuaLaTeX	
detection	<u>123</u>
file & section names	<u>227</u>
\LuaLaTeX	<u>7248</u>
LuaLaTeX (program) [requirement]	<u>45</u>
\LuaTeX	<u>7248</u>
lwrap	
loading	<u>70</u>
options	<u>70</u>
lwrap (package)	<u>70</u>
lwrap.css (file)	<u>75, 139</u>
lwrap.xdy	
customizing	<u>75</u>
lwrap.xdy (file)	<u>75, 175</u>
lwrap_formal.css (file)	<u>170</u>
lwrap_mathjax.txt (file)	<u>176</u>
lwrap_sagebrush.css (file)	<u>165</u>
lwrap_tutorial.txt (file)	<u>51</u>
lwrapmk	
customizing	<u>103</u>
[lwrapmk]:	
htmlglossary (option)	<u>86, 436</u>
printglossary (option)	<u>86, 436</u>
lwrapmk (option)	<u>115</u>
lwrapmk (program)	<u>103, 178</u>
lwrapmk.conf (file)	<u>137</u>
lwrapmk.lua (file)	<u>103</u>
\LWR@addmathjax	<u>6149</u>
\LWR@afterendverbatim	<u>4659</u>
LWR@amsmultline (boolean)	<u>335</u>
\LWR@atbeginverbatim	<u>4645</u>
\LWR@beginhideamsmath	<u>6194</u>
\LWR@blackborderpadding	<u>6907</u>
\LWR@botnavigation	<u>3308</u>
\LWR@caption@begin	<u>5722</u>
\LWR@caption@end	<u>5732</u>
\LWR@clearmidrules	<u>5093</u>
\LWR@closeparagraph	<u>3665</u>
\LWR@closeprevious	<u>3328</u>
\LWR@closetabledatatable	<u>4798</u>
LWR@colafterspec	<u>271</u>
LWR@colatspec	<u>271</u>
LWR@colbangspec	<u>271</u>
LWR@colbeforespec	<u>271</u>
\LWR@copyfile	<u>485</u>
\LWR@createautosec	<u>4050</u>
\LWR@currentcss	<u>3772</u>
\LWR@currenttextcolor	<u>7180</u>
\LWR@descitem	<u>4735</u>
\LWR@docmidrule	<u>5112</u>
LWR@doingapar (boolean)	<u>209</u>
LWR@doinghline (boolean)	<u>266</u>
LWR@doingstartpars (boolean)	<u>209</u>
LWR@doingtbrule (boolean)	<u>266</u>
\LWR@domulticolumn	<u>5186</u>

<code>\LWR@doubledollar</code>	5993	<code>LWR@infloatrow</code> (boolean)	441
<code>\LWR@endhideamsmath</code>	6202	<code>LWR@intabularmetadata</code> (boolean)	266
<code>\LWR@endminipage</code>	6769	<code>LWR@itemizeitem</code>	4700
<code>\LWR@endoffline</code>	7123	<code>LWR@latestautopage</code> (counter)	313
<code>\LWR@endsubminipage</code>	6790	<code>LWR@lateximagedepth</code> (counter)	344
<code>\LWR@ensuredoingapar</code>	3638	<code>\LWR@lateximagedepthref</code>	5518
<code>\LWR@epubprintpendingfootnotes</code>	3869	<code>LWR@lateximagenumber</code> (counter)	344
<code>LWR@externalfilecnt</code> (counter)	327	<code>\LWR@lateximagenumberref</code>	5521
<code>\LWR@filenameno blanks</code>	3889	<code>\LWR@lateximagesfile</code>	447
<code>\LWR@filestart</code>	4275	<code>LWR@Lpage</code> (counter)	344
<code>\LWR@findword</code>	379	<code>\LWR@loadafter</code>	111
<code>\LWR@floatbegin</code>	5670	<code>\LWR@loadbefore</code>	122
<code>\LWR@floatend</code>	5688	<code>\LWR@loadnever</code>	131
<code>\LWR@footnotetext</code>	3804	<code>\LWR@longtabledatacaptiontag</code>	5220
<code>\LWR@forceminwidth</code>	6896	<code>\LWR@lookforpackagename</code>	382
<code>\LWR@forcenewpage</code>	3334	<code>\LWR@LwarpEnd</code>	4394, 7284
<code>LWR@freeze thisfloat</code> (boolean)	312	<code>\LWR@LwarpStart</code>	4325, 7284
<code>\LWR@future non spacelet</code>	4775	<code>\LWR@maketitlesetup</code>	33, 4499
<code>\LWR@FVstyle</code>	17	<code>\LWR@maybe in thisfloat</code>	5705
<code>\LWR@getexparray</code>	3227	<code>\LWR@maybe new tablerow</code>	5023
<code>\LWR@getmynexttoken</code>	4782	<code>LWR@midrulecounter</code> (counter)	282
<code>\LWR@hidelatex equation</code>	6135	<code>LWR@midrules</code>	282
<code>\LWR@hline</code>	5421	<code>LWR@minipagedepth</code> (counter)	359
<code>\LWR@hspace</code>	7148	<code>LWR@minipagefullwidth</code> (boolean)	361
<code>\LWR@htmlblockcomment</code>	3517	<code>LWR@minipageheight</code> (length)	359
<code>\LWR@htmlblocktag</code>	3525	<code>\LWR@minipagestartpars</code>	7128
<code>\LWR@htmlclosecomment</code>	3489	<code>\LWR@minipagestoppars</code>	7131
<code>\LWR@htmlcomment</code>	3510	<code>LWR@minipagethispar</code> (boolean)	361
<code>\LWR@htmldivclass</code>	3554	<code>LWR@minipagewidth</code> (length)	359
<code>\LWR@htmldivclassend</code>	3557	<code>\LWR@multicolother</code>	5137
<code>\LWR@html element</code>	3567	<code>\LWR@multicolpartext</code>	5132
<code>\LWR@html element class</code>	3541	<code>\LWR@multicolskip</code>	5143
<code>\LWR@html element class end</code>	3546	<code>\LWR@multirowborder</code>	3
<code>\LWR@html element class line</code>	3560	<code>\LWR@mynexttoken</code>	4774
<code>\LWR@html element end</code>	3570	<code>\LWR@myshorttoc</code>	5787
<code>LWR@htmlfilenumber</code> (counter)	195	<code>\LWR@nameref</code>	5511
<code>\LWR@htmlfileref</code>	5514	<code>LWR@nestspan</code> (environment)	3421
<code>\LWR@htmlmathlabel</code>	6180	<code>\LWR@newhtmlfile</code>	3955
<code>\LWR@htmlmulticolumn</code>	5209	<code>\LWR@newlabel</code>	5548
<code>\LWR@htmlopencomment</code>	3489	<code>\LWR@newref</code>	5598
<code>\LWR@htmlrefsectionfilename</code>	3293	<code>LWR@nextautofloat</code> (counter)	313
<code>\LWR@HTMLsanitize</code>	6459	<code>LWR@nextautopage</code> (counter)	313
<code>\LWR@HTMLsanitizeexpand</code>	6480	<code>LWR@nextequation</code> (counter)	329
<code>\LWR@htmlsectionfilename</code>	3260	<code>\LWR@nohspace</code>	7169
<code>\LWR@htmlspan</code>	3462	<code>\LWR@nullfonts</code>	7025
<code>\LWR@htmlspanclass</code>	3470	<code>\LWR@openparagraph</code>	3643
<code>\LWR@htmltag</code>	3480	<code>\LWR@origcolspec</code>	4869
<code>\LWR@htmltagc</code>	3411	<code>\LWR@originname</code>	194
<code>\LWR@includegraphicsb</code>	100	<code>\LWR@originnames</code>	201
<code>\LWR@indexitem</code>	5941, 5945, 5949	<code>\LWR@parseaftercolumn</code>	4929

BaseJobname	115	chappg	400
HomeHTMLFilename	60, 116	chnngpage	400
HTMLFilename	60, 116	cleveref	356
IndexLanguage	67, 86, 115, 436, 540	cmapp	70
latexmk	60, 116	color	400
lwarpmk	115	comment	117
mathjax	60, 70, 115	crop	400
mathsvg	60, 70, 115	cuted	401
OSWindows	71, 113, 116	cutwin	401
warpHTML	70, 114	dblfnote	402
warpprint	70, 114	dcolumn	402
xdyFilename	76, 115	draftwatermark	403
\OSPathSymbol	28	ebook	403
OSWindows (option)	71, 113, 116	ellipsis	404
P			
packages		emptypage	404
required	122	endnotes	404
packages:		enumerate	405
a4	385	enumitem	405
a4wide	385	environ	126
a5comb	385	epigraph	406
abstract	385	eso-pic	407
addlines	388	etoolbox	112
adjmulticol	387	everyhook	125
afterpackage	124	everypage	407
afterpage	389	everyshi	408
algorithmicx	389	expl3	125
alltt	390	extramarks	408
amsmath	127	fancybox	409
amsthm	390	fancyhdr	412
anonchap	393	fancyvrb	413
anysize	393	figcaps	418
appendix	393	filecontents	125
arabicfront	394	fix2col	419
array	300	float	419
authblk	394	floatflt	420
babel	97	floatpag	421
balance	395	floatrow	421
bigdelim	395	flushend	427
bigstrut	397	fncychap	427
bookmark	397	fnpos	428
booktabs	398	fontenc	69, 428
boxedminipage	398	fontspec	69, 429
boxedminipage2e	398	footmisc	429
calc	126	footnote	430
caption	126	footnotehyper	431
caption2	399	framed	431
ccaption	399	ftnright	434
change page	399	fullpage	434
		fullwidth	434
		geometry	124, 435

getttitlestring	125	pagenote	493
glossaries	85, 435	paralist	493
graphics	351, 437	parskip	494
graphicx	351, 447	pdfscape	494
grffile	447	pdfsync	494
hyperref	307, 447	pfnote	495
hyperxmp	452	placeins	495
idxlayout	452	prelim2e	495
ifplatform	112	quotchap	496
indentfirst	453	ragged2e	497
inputenc	69, 453	realscripts	498
keyfloat	453	refcount	126
kvoptions	114	relsize	499
layout	454	romanbar	500
letltxmacro	112	romanbarpagenumber	500
letterspace	455	rotating	501
lettrine	455	rotfloat	501
lips	456	savetrees	502
listings	457	scalefnt	502
lmodern	69	sectsty	502
longtable	460	setspace	503
lscape	462	shadow	504
ltxcaption	462	showidx	505
lwrap	70	showkeys	505
makeidx	126	sidecap	505
marginfit	463	sidenotes	506
marginfix	463	siunitx	89, 351, 508
marginnote	463	soul	508
mcaption	464	stabular	510
mdframed	464	subfig	510
memoir	552	supertabular	516
metalogo	473	syntonly	517
microtype	70, 123, 473	tabls	518
midfloat	474	tabularx	518
moreverb	474	tabulary	519
mparhack	476	textarea	519
multicol	476	textcomp	70, 520
multirow	477	textpos	523
nameref	480	theorem	523
needspace	480	threeparttable	527
newclude	97, 480	tikz	528
newfloat	126	titlesps	529
newtxmath	89	titleref	532
newunicodechar	69, 481	titlesec	532
nextpage	481	titletoc	534
nicefrac	89	titling	535
nonumonpart	481	tocbibind	87, 540
nopageno	482	tocloft	542
nowidow	482	transparent	547
ntheorem	482	trivfloat	548

typearea	549	prelim2e (package)	495
ulem	550	\printauthor	245, 4466, 4485
units	89	\printdate	245, 4477, 4487
upref	551	printglossary (option) [lwarpmk]	86, 436
verse	551, 552	\printindex	5924
wallpaper	553	\PrintStack	3313
wrapfig	554	\printthanks	244, 247
xcolor	555	\printtitle	245, 4450, 4484
xfrac	563	problems	105
xifthen	125	programs	
xltxtra	565	utility	44
xmpinl	566	programs:	
xparse	124	[requirement]:	
xstring	125, 126	LuaLaTeX	45
xtab	566	MathJax	45
zref	126	pdfcrop	45
zwpagelayout	568	pdfLaTeX	45
page		pdfseparate	45, 49
inaccessible	61	pdftocairo	45, 49
\pagebreak	7175	pdftotext	45, 49
\pagecolor	43	perl	50
pagenote (package)	493	XeLaTeX	45
\pagenumbering	3345	Adobe	43
\pageref	5605	AsciiDoc	42
\pagerefPageFor	5604	AsciiDoctor	42
\pagestyle	3337	Asciidoctor-LaTeX	42
Pandoc (program)	42	Flare	43
\paragraph	4250	FrameMaker	43
paralist (package)	493	GELLMU	41
\parbox	6861	GladTeX	42
\parsemulticolumnalignment	5173	Hevea	41
parskip (package)	494	InDesign	43
\part	4201	LaTeX2HTML	41
pdfcrop (program) [requirement]	45	LaTeXXML	41
pdfLaTeX (program) [requirement]	45	LibreOffice	42
pdfscape (package)	494	Linux	71, 113
pdfseparate (program) [requirement]	45, 49	lwarpmk	103, 178
	45, 49	Mac OS	71, 113
pdfsync (package)	494	Madcap	43
pdftocairo (program) [requirement]	45, 49	MathJax	88, 326
pdftotext (program) [requirement]	45, 49	MS-Windows	71, 113
Perl	50	OpenOffice	42
perl (program) [requirement]	50	Pandoc	42
pfnote (package)	495	Plastex	41
\phantomsection	7226	TeX2page	41
picture (environment)	358, 6760	TeX4ht	41
placeins (package)	495	TtH	41
Plastex (program)	41	Unix	71, 113
\popclose	3195	Windows	71, 113
Poppler	45, 49	Word	42

mathsvg option	115	threeparttable (package)	527
syntonly (package)	517	tikz	
T			
<code>\tableofcontents</code>	63	catcodes	528
tabls (package)	518	dollar redefined	528
tabular		tikz (package)	528
baseline	275	tikzpicture (environment)	8
HTML columnn conversion	276	<code>\title</code>	73
macros inside	93, 264	titlepage	
misplaced <code>\noalign</code>	93, 264	subtitle and published	252
misplaced alignment tab character &		titlepage (environment)	73, 4434
.....	92, 263	titles (package)	529
multicolumn with <code>multirow</code>	479	titleref (package)	532
row corruption	93, 264	titles (option) [tocloft]	84
<code>\TabularMacro</code>	5341	titlesec (package)	532
tabularx (package)	518	titletoc (package)	534
tabulary (package)	519	titling (package)	535
tagged equations		titlingpage (environment)	14, 73
Mathjax	88, 326	[tocbibind]:	
<code>\TeX</code>	7231	numindex (option)	87, 541
<code>TeX2page</code> (program)	41	tocbibind (package)	87, 540
<code>TeX4ht</code> (program)	41	tocdepth (counter)	60
textarea (package)	519	[tocloft]:	
<code>\textbf</code>	6987	titles (option)	84
<code>\textcolor</code>	33	tocloft (package)	542
textcomp (package)	70, 520	<code>\tracinglwrap</code>	107, 505
<code>\textgreater</code>	3241	transparent (package)	547
<code>\textit</code>	7002	trivfloat	
<code>\textless</code>	3241	with newfloat, algorithmic	549
<code>\textmd</code>	6984	trivfloat (package)	548
<code>\textnormal</code>	7011	troubleshooting	105
textpos (package)	523	HTML debug comments	132
<code>\textrm</code>	6990	<code>\ttfamily</code>	7061
<code>\textsc</code>	7005	TtH (program)	41
<code>\textsf</code>	6993	tutorial.tex (file)	51
<code>\textsl</code>	7008	typearea (package)	549
<code>\textsubscript</code>	7069	U	
<code>\textsuperscript</code>	7068	ulem (package)	550
<code>\texttt</code>	6996	<code>\underline</code>	7072
<code>\textup</code>	6999	underscore	
<code>\tfl@chapter@fix</code>	548	filename	60, 77
<code>\thanks</code>	74	Unicode	
<code>\thanksmarkseries</code>	84	enhanced coverage	69
<code>\theauthor</code>	245	file & section names	227
<code>\thedata</code>	245	input characters	111
theindex (environment)	5931	selection	69
theorem (package)	523	units (package)	89
<code>\thetitle</code>	245	Unix (program)	71, 113
<code>\thispagestyle</code>	3338	<code>\up</code>	7070
		upref (package)	551

